SANDIA REPORT

SAND88—1432 • UC—32
Unlimited Release
Printed June 1989

BLOT – A Mesh and Curve Plot Program for the Output of a Finite Element Analysis

Amy P. Gilkey, John H. Glick

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A07
Microfiche copy: A01

SAND88-1432
Unlimited Release
Printed June 1989

Distribution
UC-32

BLOT – A Mesh and Curve Plot Program for the Output of a Finite Element Analysis

Amy P. Gilkey
Applied Mechanics Division III
John H. Glick
Fluid Mechanics and Heat Transfer Division I
Sandia National Laboratories
Albuquerque, New Mexico 87185

Abstract

BLOT is a graphics program for post-processing of finite element analysis output that is presented in the EXODUS database format. It is command driven with free-format input and can drive any graphics device supported by the Sandia Virtual Device Interface.

BLOT produces mesh plots with various representations of the analysis output variables. The major mesh plot capabilities are deformed mesh plots, line contours, banded contours, vector plots of two or three variables (e.g., velocity vectors), and symbol plots of scalar variables (e.g., temperature). Pathlines of analysis variables can also be drawn on the mesh. BLOT's features include element selection by material, element birth and death, multiple views for combining several displays on each plot, symmetry mirroring, and node and element numbering.

BLOT can also produce X-Y curve plots of the analysis variables. BLOT generates time-versus-variable plots or variable-versus-variable plots. It also generates distance-versus-variable plots at selected time steps where the distance is the accumulated distance between pairs of nodes or element centers.

ACKNOWLEDGEMENTS

This program incorporates several existing programs: DETOUR [1], TPLOT [2], SPLOT [3], and GROPE [4].

The first version of DETOUR was written by Dennis P. Flanagan. The first version of TPLOT was written by Zelma E. Beisinger and Charles M. Stone [5]. The first version of SPLOT was written by Mary R. Sagartz [6]. The code for the neutral file output was written by Greg D. Sjaardema. The spectrum color scale algorithm was provided by Johnny H. Biffle.

Contents

Tables

Figures

1. Introduction

Post-processing is an important part of the engineering analysis process because it helps the analyst to interpret the large amounts of data produced by the analysis code. Post-processing capabilities include the selective plotting and printing of the mesh description data and solution data for a problem.

BLOT is a post-processing program developed in the Engineering Analysis Department at Sandia National Laboratories. The database read by BLOT must be in the EXODUS format [7]. An EXODUS database contains the mesh description and, optionally, results data from an analysis. Results data consists of the values of the database variables at one or more times (time steps). Section 1.1.1 summarizes the EXODUS format.

BLOT is the combination of several distinct subprograms. When a BLOT session is started, the user is placed in what is called the BLOT level. From here, the user can access any of the subprograms described below. Some of them must be entered explicitly (by typing the subprogram name), and some are accessed implicitly by typing commands that it recognizes from the BLOT level or from another subprogram.

- MESH plots the undeformed mesh. It is accessed implicitly from either the TPLOT or SPLOT subprograms.
- DETOUR plots the deformed mesh at user selected database times. It will also represent output variable values on the deformed mesh with line or banded contours, and vectors. DETOUR must be explicitly entered by typing its name from the BLOT level or from another subprogram.
- PATHLINE plots pathlines on the undeformed mesh. The locations of the points which are connected to create a pathline are stored in two (three, if a 3D database) variables. PATHLINE must be explicitly entered by typing its name from the BLOT level or from another subprogram.
- TPLOT creates history plots and variable-versus-variable plots. Histories plot the value of a variable versus time. Variable-versus-variable plots display the value of one variable versus the value of another variable over a user selected time range. TPLOT must be explicitly entered by typing its name from the BLOT level or from another subprogram.
- SPLOT creates profile plots where a database variable is plotted against distance along a user specified spatial profile. SPLOT must be explicitly entered by typing

- GROPE allows the user to selectively examine the data in the EXODUS database. This includes the mesh description, the variable names, the times for which data exist, and the solution data at each time. The user can print the data to the screen or to a print file. GROPE is accessed implicitly by typing commands that it recognizes from the BLOT level or any other subprogram.

1.1 Input to BLOT

The input to BLOT is a database file in the EXODUS format and a series of user supplied commands.

1.1.1 EXODUS Files

The EXODUS database format is described briefly in Appendix A and in detail in [7]. This section discusses some of the features mentioned in this document.

The first part of the EXODUS database describes the mesh. This part of the database is the GENESIS database format [8] that also serves as the input to a finite element analysis. The mesh description includes the nodal coordinates and the element connectivity. BLOT can process a GENESIS database to generate undeformed mesh plots.

Each element in the database is assigned to an "element block". An element block distinguishes a material or an element type (such as a truss or quadrilateral). A specific element variable may be undefined for some element blocks, meaning that the variable has no value for elements in that element block.

The GENESIS database also may contain "node sets" and "side sets". A node set is a collection of nodes that the user has defined. A side set is a user-defined collection of element sides described by an element number and the nodes in the side. BLOT can display either type of set on the mesh.

Following the mesh description is information about the database variables, then a series of time steps. Each variable represents a result of the analysis at each time step. The EXODUS database has a user-defined name associated with each variable. This name is used to reference the variable and to label any plot involving the variable. A time step consists of a time and the values for all the variables at that time.

The EXODUS format defines four types of variables:

- A history variable has a value representative of the system as a whole at each time step (e.g., the total energy).

- A global variable is the same as a history variable except that global variables are only included in "whole" time steps (explained below).
- A nodal variable has a value for every node of the mesh at each whole time step in the analysis (e.g., the displacement in the x-direction).
- An element variable has a value for every element of the finite element analysis at each whole time step (e.g., the stress in the x-direction). An element variable may be undefined for some element blocks, meaning that the variable has no value for elements in that element block.

There are two types of time steps in an EXODUS database: a "history-only" time step contains the values for the history variables only; a "whole" time step includes the values for all the variables (history, global, nodal, and element).

An EXODUS database variable naming convention dealing with mesh deformation is used by BLOT. One of the capabilities of the DETOUR subprogram is to plot the deformed mesh at user specified database times. This requires the existence of a set of nodal displacement variables that define the displacement of each node from its original location.

The displacement variables are the first two or three nodal variables (one for each coordinate) defined in the EXODUS database if and only if these variables begin with the letter "D" and end with the last character of the coordinate name. For example, if the coordinate name is "X", nodal variable names "DX" and "DISPX" are valid displacement variable names. If the first nodal variables are not valid displacement variables, a warning is displayed and only the undeformed mesh can be plotted.

This convention of specifying displacement variables is not part of the EXODUS database definition, but must be observed by users of the BLOT to generate deformed mesh plots.

1.1.2 BLOT Commands

The user controls the actions of BLOT by specifying a series of commands. A command may consist of the command name alone, or the command name followed by a number of fields, each containing a keyword or parameter value. The beginning of Chapter 5 gives general rules on typing BLOT commands. The commands used to control BLOT are discussed in Chapters 2 through 4. These chapters deal only with the functionality of a particular BLOT capability. The specific syntax of the commands is given in Chapter 5. Appendix D is a summary of all BLOT commands and can be used as a reference for experienced users.

1.2 Output from BLOT

The following information is generated by BLOT in the course of a typical session:

- Plot(s) and/or a plot file – Plots generated by BLOT can be displayed on a terminal or written to a plot file for later display on a hardcopy device. Which of these happens depends on the user's device specification when BLOT is executed. If only a terminal device was specified, all plots are displayed on that terminal. Similarly, if only a hardcopy device is specified, all plots are written to a plot file. If both a terminal device and a hardcopy device are specified, a requested plot can be displayed on the terminal, printed to the plot file, or both.

  Not all systems support having both a terminal device and a hardcopy device active during a session. Appendix F contains site supplements for supported operating systems. These show how to specify devices and if the system supports multiple devices. The valid devices are those supported by the Sandia Virtual Device Interface [10].

- Print file – This file contains any database information routed to it by the user.

- Log file – This optionally created file traces the BLOT session. Each correct command that the user enters is written to the log file.

2. General Use of BLOT

2.1 Executing and Interacting with BLOT

BLOT is executed by typing a system-dependent execution command that specifies a number of parameters:

- The EXODUS database to be post-processed.
- The graphics device(s) to which plots will be printed. Section 2.2 discusses graphics devices in BLOT.
- An optionally specified file containing BLOT commands (the startup command file).
- A batch flag indicating if the BLOT session is to be interactive or batch.

The system manager should know the format of this command. The commands for executing BLOT on currently supported systems are described in Appendix F.

Whether BLOT is run interactively or in batch, it operates by reading user supplied commands and taking appropriate actions. BLOT can read commands from the terminal (if an interactive session) or a file.

For a batch session, the user must have specified a startup command file from which BLOT reads all commands. For an interactive session in which the user has specified a startup command file, BLOT first processes commands from this file. If a command in this file does not terminate the session, BLOT then prompts for further commands from the terminal. For an interactive session in which the user has not specified a startup command file, BLOT begins by prompting for commands at the terminal.

Using the CMDFILE command, the user can at any time during a session redirect the source of commands to a new file. BLOT processes commands from this file until it is exhausted, and then returns to its previous source for the next command. Since the CMDFILE command can itself appear in a file of commands, the user has great flexibility in using command files written for specific tasks.

The HELP command gives the user access to a description of all valid BLOT commands.

The LOG command requests that the log file be saved when BLOT is exited. Each correct command that the user enters during a session (excluding the SHOW, HELP, SELECT, LIST, and PRINT commands) is echoed to the log file.

The EXIT command causes the current BLOT session to terminate.

These and other program control and information commands are described in Section 5.1.1.

2.2 Graphics Devices

BLOT can drive both a graphics terminal device and a hardcopy (file) device during a single session. This allows the user to generate plots on a graphics terminal and then copy selected plots to a hardcopy device. The capability of supporting multiple devices during a single session may not be available on all systems. The site supplements in Appendix F have information on this for supported systems.

The first graphics device specified on the execution command line is the primary device. This must be specified. The second device specified is the hardcopy device and is optional. If only a primary device is specified, it can be either a terminal or hardcopy device. If both a primary and hardcopy device are specified, the primary device must be a terminal device and the second device specified must be a hardcopy device.

BLOT can drive any graphics device supported by the Sandia Virtual Device Interface (SVDI) [10].

2.3 Plot Sets

BLOT operates in a command/plot cycle. The user first issues commands that define the plot or plots desired. These plots are referred to as the plot set. The PLOT command is then used to print the plots defined by the plot set. Each plot is printed to the primary graphics device. If this device is a terminal display, then at the completion of each plot, BLOT waits for a user response. The user can 1) display the next plot in the plot set, 2) add text to the plot, 3) terminate the display of the plot set, or 4) copy the plot to the plot file (only if a hardcopy device was also specified). If the primary graphics device is a hardcopy device, the entire plot set is copied to the plot file with no intervening user interaction. Using the AUTO command, the user can inhibit the prompting at the completion of each plot. In this case, the entire plot set is printed to the primary device with no user intervention.

If both primary and hardcopy devices were specified on the BLOT command line, the HARDCOPY command can be used. This prints the current plot set to the plot file without first displaying it on the terminal display.

The formats for the commands discussed in this section are described in Section 5.1.5.

2.4 Color Scales

BLOT makes use of two color scales in making plots: the standard color scale consists of up to six colors. The COLOR command is used to specify the number of colors in the standard color scale. The spectrum color scale consists of up to 248 colors, and the SPECTRUM command is used to specify the number of colors defining it. This may be zero, in which case the standard color scale is used in place of the spectrum color scale. Section 5.1.4 discusses the COLOR and SPECTRUM commands.

In using color to make plots, BLOT uses either the standard color scale or the spectrum color scale, depending on the situation. When color is to be used, this manual will always specify which color scale applies.

2.5 Time Step Selection

In defining a plot set, the user most often will want to define the database time steps from which data will be drawn. Time step selection is performed in one of the following modes:

- **Interval-Times Mode** selects time steps at uniform intervals between a minimum and a maximum time. The user specifies the minimum and maximum times and either the time interval or the number of times to be selected. The first selected time can be the minimum time or the minimum time plus the interval.

- **All-Available-Times Mode** selects all time steps between a minimum and a maximum time.

- **User-Selected-Times Mode** selects time steps which are explicitly specified by the user.

The nearest time step from the database is chosen for each selected time.

"History-only" time steps are chosen only if the current subprogram can process history variables.

Section 5.1.3 lists and discusses the commands used for time step selection.

2.6 Database Listing and Printing (GROPE)

The GROPE subprogram has the capability of listing to the screen or printing to the print file selected data from the EXODUS database. The SELECT command allows the user to specify the portion of the database to be listed or printed. The LIST command

displays selected items from the database to the screen. The PRINT command prints the selected items from the database to the print file. These commands are valid from the BLOT level or from any subprogram. They are described in detail in Section 5.1.2.

2.7 General Graphics Options

There are a number of graphics options that are common to all plots, whether they are 2D or X-Y plots. These include whether to display QA information on the plot legend, whether to number the plot axes, and whether the plot legend is displayed on the plot. The formats for these and other general graphics options are discussed in Section 5.1.4.

3. 2D Plots

2D plots display the mesh defined in the EXODUS database. Database variable values may be represented on them.

There are several different types of 2D plots:

- The undeformed wireframe mesh is displayed by the MESH subprogram. Section 3.3 describes this capability.
- The deformed mesh is plotted by the DETOUR subprogram. The mesh may be plotted as a wireframe drawing or as a painted drawing in which each element is painted rather than outlined. Database variable values may be represented on the wireframe mesh as lined or banded contours, vectors, or special element symbols. DETOUR capabilities are described in Section 3.4.
- Pathlines defined by selected variables may be plotted on the undeformed wireframe mesh using the PATHLINE subprogram. This capability is described in Section 3.5.

3.1 Mesh Display

All 2D plots display the mesh, which is defined by three types of lines:

- mesh boundary lines, which form the boundary of the mesh (the problem domain).
- element block boundary lines, which form the boundary between element blocks.
- element boundary lines, which form the boundary between individual elements.

By default, mesh boundary lines will be the thickest on the plot and will be drawn in the foreground color. The foreground color is initially set to white and can be changed with the FOREGROUND command. Element block boundary lines will be slightly thinner and will also be drawn in the foreground color. Element boundary lines are the thinnest and are normally drawn in the color assigned to its element block. In certain DETOUR plots they will drawn in black to contrast them from other colors on the plot (see Section 3.4.2). Each element block is assigned a default color from the standard color scale by BLOT, although the BLKCOL command can be used to change these.

It is also possible to change the thickness assigned to the three types of lines with the LINETHICKNESS command. See Section 5.2.4 for details on these commands.

Nodes and/or element numbers can be printed on the mesh. Node numbers are printed in the foreground color to the upper right of the node. Element numbers are printed in the color assigned to its element block at the center of the element. The NUMBER command (Section 5.2.1) controls whether numbering is performed.

Node sets and side sets can also be displayed on the mesh. If node sets are displayed, each node in a selected set is marked with an X in the color assigned to the node set. The identifiers of selected sets appear in the plot legend, also in the color assigned to the set. If side sets are displayed, the nodes that make up a side in a selected set are connected by lines in the color assigned to the side set. The first two nodes displayed for each side are connected with an arrow to indicate the node ordering within the set. The identifiers of selected sets appear in the plot legend in the color assigned to the set. The NSETS and SSETS commands (Section 5.2.1) control the display of node sets and side sets.

3.2 General 2D Plot Control

Several mesh display control capabilities are common to all the 2D plot subprograms.

3.2.1 Multiple Views

A single plot may provide up to four "views" of a mesh. The views may be copies of the original mesh to show different representations of the mesh (for example, the deformed mesh versus the undeformed mesh). The views may also be a reflection of the mesh around a symmetry axis, providing a way to display a whole object from a symmetric mesh that is a quarter or half of that object. Symmetric views are not allowed for a 3D mesh.

The plot parameters that may vary between views are limited. The command descriptions in Chapter 5 will state if a command is view dependent.

Each view has a number associated with it depending on how the views are defined. The view numbering for each possible combination of views is shown in Figure 3.1.

Normally the default mesh window for each view is square. However, if there are two views and the non-divided axis length (derived from the default or user-defined mesh limits) is at least twice the divided axis length, the views are rectangular (filling a square). The window may be changed to an arbitrary rectangle with the SQUARE OFF command.

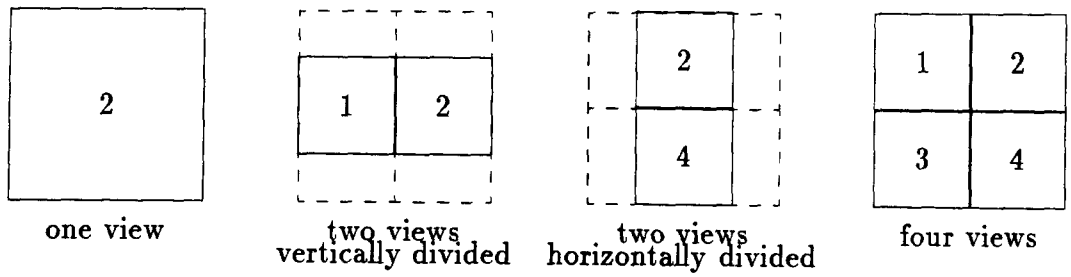

Figure 3.1. View numbering for multiple views.

The commands associated with controlling multiple views are detailed in Section 5.2.3.

3.2.2 Mesh Control

A number of commands are available to the user for controlling mesh display parameters. Some of these parameters are view dependent; that is, they can have a different setting for each view. These include display of node and element numbers, display of node sets and side sets, and whether to display all mesh lines or only the mesh boundary and element block boundaries. The view-dependent commands are detailed in Section 5.2.1.

Some of the mesh control parameters cannot vary between views. These include the shape of the mesh window, the region of the mesh to be displayed, the color to be assigned to each element block, and the thickness with which to draw the lines of the mesh. The view-independent commands are detailed in Section 5.2.4.

3.2.3 Active Element Selection

The user may select/deselect elements by element block. This provides a way to make meaningful contour plots if the value of a variable to be contoured changes greatly across an element block boundary (e.g., stress in foam versus stress in steel). Elements are allowed to "be born" (to be selected) and "die" (to be deselected) between time steps on an individual basis.

Elements are selected in two ways: by element block and by a special birth/death flag. "Active" elements must be selected by both criteria.

One or more element blocks may be "selected". Only elements in a selected element block contribute to any variable plot (e.g., a contour plot). On a mesh plot, lines

separating elements in the same element block (in some display modes) are only drawn for elements in a selected element block.

Element variables may be undefined for some element blocks. If so, only the elements in an element block for which the variable is defined contribute to a plot involving that variable.

Elements are allowed to be born or to die. If an element is "dead", it is ignored in all calculations and plots, including a simple mesh plot. The user specifies an element variable which gives the state of the element for each time step: the element is alive if the value is zero, dead if it is not zero. This capability can be turned on or off upon command.

The commands used for selecting active elements are detailed in Section 5.2.2.

3.2.4 3D Mesh Display

A 3D mesh can be displayed in any subprogram. Only the surface of the mesh may be displayed. The surface is defined as an element face which has no matching face. All functions of database variables apply only to the surface of the mesh. The node and element numbers and the node sets and side sets are also displayed only on the surface of the mesh. The one exception is the pathlines which may go through the interior of the mesh.

The hidden line algorithm is not perfect and may not work on all element types. In particular it has trouble with determining the "edge" of the mesh.

The axes displayed for a 2D mesh represent the coordinates of the mesh. In 3D the axis values have little resemblance to the original mesh coordinates. The axis values are only useful for selecting a position on the displayed mesh. Since it is much simpler to use a graphics cursor, the axis values are left off the 3D mesh plot by default.

The commands for controlling the view of 3D meshes are detailed in Section 5.2.5.

3.3 MESH Subprogram

The MESH subprogram is used to display the undeformed wireframe mesh. It is accessed by typing the MESH command from either of the X-Y curve subprograms (SPLOT or TPLOT). This will plot the mesh without leaving the subprogram from which it was executed. The MESH command is described in Section 5.5.5.

3.4 DETOUR Subprogram

Plots displaying the deformed mesh are generated by the DETOUR subprogram. The deformed mesh may be displayed alone or representations of the database variable values may be plotted on the mesh. These representations may be lined or painted contours, vectors, or special symbols. These different types of plots are discussed below in the section on the display modes of DETOUR.

Whatever the type of plot to be generated, the database time steps for which the plots are to be made must be selected. By default, all time steps are selected. Section 2.5 discusses the selection of time steps.

3.4.1 Deformed Mesh Display

The display of the deformed mesh is common to all the modes of DETOUR. All of the descriptions and commands related to mesh plots discussed in Sections 3.1 and 3.2 apply to deformed mesh plots.

To plot a deformed mesh, DETOUR adjusts the location of each node using the displacement variables and a magnifying factor. Displacement variables (described in Section 1.1.1) must be present in the EXODUS database to plot the deformed mesh. Otherwise, the undeformed mesh will always be plotted. The deformed mesh location of a node is calculated by taking the displacement values (at the current time step) for each coordinate direction, multiplying them by the magnification factor, and adding each to the original nodal coordinate value. The default magnification factor is 1.0 for both 2D and 3D meshes. The magnification factor may be set using the MAGNIFY command (Section 5.2.4).

3.4.2 DETOUR Display Modes

DETOUR operates in several "display modes", which are described below. The DETOUR commands used for entering these modes are described in Section 5.3.1.

- **Wireframe Mesh Mode** plots a wireframe drawing of the deformed mesh. This mode is entered by issuing the WIREFRAME command. This is also the initial display mode when DETOUR is entered.
- **Solid Mesh Mode** paints the deformed mesh, using a different color for each element block. DETOUR will assign default colors to each element block, or the BLKCOL command can be used to assign colors to them. In these plots, the element boundary lines are drawn in black. This mode is entered by typing the SOLID command.

- **Line Contour Mode** plots contours of a variable on the mesh. The CONTOUR command is used to both enter line contour mode and to specify the variable to be contoured. This may be either a nodal or element database variable. Specifying values for each contour is done using the commands described in Section 5.3.2. A mapping is then made between the contour values and the colors in the spectrum color scale. On the plot, each contour is plotted in its assigned color and is also labeled with a letter. The value corresponding to each color and letter is listed in the plot legend.
- **Paint Contour Mode** paints contours of a variable on the mesh. The PAINT command is used to both enter paint contour mode and to specify the variable to be contoured. As with line contours, this may be either a nodal or element database variable. Assignments of values to contours is done in the same manner as for line contours. With painted contours, however, each color from the spectrum color scale is assigned the values between two adjacent contour values. The value range corresponding to each color is shown in the plot legend.
- **Vector Mode** plots a two- or three-variable vector (depending on the dimension of the mesh) for each node or element on the mesh. The VECTOR command is used to both enter vector mode and to specify the variables defining the vectors to be plotted. The variables must be either all nodal type or all element type. If nodal variables are specified, vectors are drawn at each node in the color of the node. If element variables are specified, vectors are drawn at the center of the element in the color of the element block to which the element belongs. The VSCALE command (Section 5.3.3) may be used to control the relative length of the vectors.
- **Element Symbol Mode** plots a symbol for each element on the mesh. The SYMBOL command is used to enter the element symbol mode and to specify an element variable that defines the symbol to be plotted. A symbol is plotted at the center of each element. The symbol may be a letter; an "X" indicating a negative variable value, or an "O" indicating a positive value. The symbol may also be a line oriented at the angle specifed by the variable value. The VSCALE command is used to control the size of the symbol.

3.5 PATHLINE Subprogram

Pathlines may be plotted on the undeformed wireframe mesh using the PATHLINE subprogram. Each pathline is defined by a set of variables (two for a 2D mesh and three for a 3D mesh). The variables may be a set of history or global variables or a set of nodal variables at specified nodes. Each time step in the database defines a point on the pathline, and each of the variables gives a coordinate of that point.

Multiple pathlines may be defined for an individual plot. The PATHLINE subprogram commands are described in Section 5.4.

4. X-Y Curve Plots

The X-Y curve plotting capability consists of two subprograms: TPLOT and SPLOT. TPLOT plots time histories of variables and variable-versus-variable plots. SPLOT plots variable values as a function of mesh position.

A number of attributes common to all X-Y curve plots can be controlled:

Axis labeling and scaling are controlled using commands described in Section 5.5.2. BLOT will assign default axis labels and scales if they are not explicitly assigned.

The display of individual curves on a plot is controlled using the commands described in Section 5.5.3. Some of the these curve attributes are:

- Each curve in the current plot set may be drawn on its own plot, or multiple curves may be overlaid on a single plot.
- BLOT assigns each curve that appears on a plot a color from the spectrum color scale. As discussed in Section 2.4, the user has control over the number of colors in the spectrum color scale.
- The user can specify if the points defining each curve on a plot are to be connected with a straight line, and if so, are all to be solid or does BLOT cycle through six line types (solid, dotted, dot dash, short dash, long dash, and medium dash). If the line type varies between curves, it is included in the legend.
- The user can control whether symbols will mark the points on each curve. If so, symbols can be assigned to each curve, or BLOT can assign them. If the symbols vary between curves, they are included in the legend.

Each curve on a plot is identified in the legend. The curves on a plot are numbered if they cannot be distinguished by their line type and symbol. Thus, if the line type and symbol are constant or if more than six curves are on a plot, all curves are numbered. These numbers are then included in the legend.

The curve data can be written in "neutral file" format. This file can then be read by the GRAFAID program [9]. GRAFAID is an interactive X-Y plot program developed for use on VAX computers under the VAX/VMS operating system. It provides an interactive capability for plotting and comparing multiple sets of X-Y data. It has numerous utilities for X-Y data analysis, manipulation, and creation. Utilities are provided for appending X-Y data sets, performing calculations involving X-Y data, editing data, enveloping data, calculating FFT's, integrating and differentiating data,

windowing data, and doing statistical calculations. The commands used for generating a neutral file are described in Section 5.5.4.

4.1 TPLOT Subprogram

TPLOT generates time history plots and variable-versus-variable plots. Both types of plots need a set of user-selected database time steps which specify the times at which the plot values are taken. A time history curve displays the value of a variable at a time versus the time. A variable-versus-variable curve displays the value of one variable at a time versus the value of another variable at the same time.

Any type of database variable may be plotted. For a nodal or element variable, TPLOT plots values for a selected node or element. The user specifies the node or element number with the variable. For example, the user can plot load versus displacement for any node, stress versus strain from any element, or global values of kinetic energy versus time.

The TPLOT plot set is the set of all defined curves for all requested variables. The curves can all be overlaid on one plot or plotted one curve to a plot.

The commands used for specifying curves in TPLOT are described in Section 5.6.

4.2 SPLOT Subprogram

SPLOT plots the analysis results as a function of position in a body (for example, the value of the displacement around the circumference of a cylinder). A distance parameter is created by connecting the distances between user-selected pairs of nodes or pairs of element centers. SPLOT plots the value from the node or element center versus the accumulated distance at that node or element center.

The user first selects the nodes or elements, then the nodal or element variable. SPLOT generates plots of cumulative distances between the selected nodes versus the values of the nodal variable at those nodes or the cumulative distances between the centers of the selected elements versus the values of the element variable at those elements. A curve is generated for the variable values at each user-selected database time step.

The SPLOT plot set is the set of all defined curves for all requested variables and all selected times. The curves for a plot set can be plotted one curve to a plot or can be overlaid on a plot in one of two ways: the curves for all requested variables can be generated on one plot for each time step or the curves for all times steps can be generated on one plot for each variable.

The commands used for specifying curves in SPLOT are described in Section 5.7.

5. Command Formats

BLOT prompts for commands from the user with the name of the current subprogram. The program starts at the BLOT level. At this level the user can only request general information and enter a subprogram.

The commands are in free-format and must adhere to the following syntax rules.

- Valid delimiters are a comma or one or more blanks.
- Either lowercase or uppercase letters are acceptable, but lowercase letters are converted to uppercase except in user-defined text that appears on a plot (such as the plot caption).
- A "$" character in any command line starts a comment. The "$" and any characters following it on the line are ignored.
- A command may be continued over several lines with an ">" character. The ">" and any characters following it on the current line are ignored and the next line is appended to the current line.

Each command has an action keyword or "verb" followed by a variable number of parameters.

The command verb is a character string. It may be abbreviated, as long as enough characters are given to distinguish it from other commands.

The meaning and type of the parameters is dependent on the command verb. Most command parameters are optional. If an optional parameter field is blank, a command-dependent default value is supplied. Below is a description of the valid entries for parameters.

- A numeric parameter may be a real number or an integer. A real number may be in any legal FORTRAN numeric format (e.g., 1, 0.2, -1E-2). An integer parameter may be in any legal integer format.
- A string parameter is a literal character string. Most string parameters may be abbreviated.
- Variable names must be fully specified. The blank delimiter creates a problem with database variable names with embedded blanks. The program handles this by deleting all embedded blanks from the input database names. For example, the variable name "SIG R" must be entered as "SIGR". The blank must be deleted in

any references to the variable. All database names appear in uppercase without the embedded blanks in all displays.

- Screen and mesh positions may be selected with the graphics cursor (also known as the graphics locator). Cursor input is device-dependent and uses the VDI graphics locator routines. When the program prompts for the position, the user positions the graphics cursor (e.g., the crosshairs) on the screen, then selects the position by pressing any printable keyboard character (e.g., the space bar).
- Several parameters allow a range of values. A range is in one of the following forms:
  - "n_1" selects value n_1,
  - "n_1 TO n_2" selects all values from n_1 to n_2,
  - "n_1 TO n_2 BY n_3" selects all values from n_1 to n_2 stepping by n_3, where n_3 may be positive or negative.

  If the upper limit of the range is greater than the maximum allowable value, the upper limit is changed to the maximum without a message.

The notation conventions used in the command descriptions are:

- The command verb is in **bold** type.
- A literal string is in all uppercase SANSERIF type and should be entered as shown (or abbreviated).
- The value of a parameter is represented by the parameter name in *italics*.
- A literal string in square brackets ("[ ]") represents a parameter option which is omitted entirely (including any following comma) if not appropriate. These parameters are distinct from most parameters in that they do not require a comma as a place holder to request the default value.
- The default value of a parameter is in angle brackets ("< >"). The initial value of a parameter set by a command is usually the default parameter value. If not, the initial setting is given with the default or in the command description.

Each command section header tells which subprogram must be active for the commands in the section. The command may behave differently depending on the current subprogram. These differences are documented in the command description under a heading of "Issued from *subprogram*".

After each command is entered, the effect of the command is displayed. For example, after a TMIN command, the time step parameters are displayed.

The current setting of a command parameter can be displayed with the SHOW command. The LIST command displays database information that may be needed in some commands (e.g., the variable names and the time step times).

5.1 General Commands (any subprogram)

5.1.1 Program Control and Information (any subprogram, including BLOT level)

LOG

LOG requests that the log file be saved when the program is exited. Each correct command that the user enters (excluding the SHOW, HELP, SELECT, LIST, and PRINT commands) is written to the log file.

RESET

RESET resets the plot parameters for the current subprogram to their initial values. All parameters are reset unless otherwise noted in the description of the command that sets the parameter. Some graphics options (see Section 5.1.4) and neutral file options (see Section 5.5.4) are not reset.

EXIT

The EXIT command exits immediately from the program. This is the only safe way to exit the program.

DETOUR

DETOUR switches to the DETOUR subprogram. All mesh plot parameters are reset (if the current subprogram is not DETOUR) unless otherwise noted in the description of the command that sets the parameter.

This command may not be abbreviated except from the BLOT level.

PATHLINE

PATHLINE switches to the PATHLINE subprogram. All mesh plot parameters are reset (if the current subprogram is not PATHLINE) unless otherwise noted in the description of the command that sets the parameter.

This command may not be abbreviated except from the BLOT level.

TPLOT

TPLOT switches to the TPLOT subprogram and resets all X-Y curve plot parameters (if the current subprogram is not TPLOT) unless otherwise noted in the description of the command that sets the parameter.

The mesh plot parameters are reset if the current subprogram is not an X-Y curve subprogram. This effects all parameters unless otherwise noted in the description of the command that sets the parameter.

This command may not be abbreviated except from the BLOT level.

SPLOT

SPLOT switches to the SPLOT subprogram and resets all X-Y curve plot parameters (if the current subprogram is not SPLOT) unless otherwise noted in the description of the command that sets the parameter.

The mesh plot parameters are reset if the current subprogram is not an X-Y curve subprogram. This effects all parameters unless otherwise noted in the description of the command that sets the parameter.

This command may not be abbreviated except from the BLOT level.

CMDFILE *file_name* <no default>

CMDFILE causes BLOT instructions to be read from the file specified by *file_name*. Instructions are read from this file until one of the two following conditions are met:

a) An end-of-file (EOF) is detected in the file. In this case, BLOT returns to the source that issued the CMDFILE instruction for the next instruction. If this was the terminal, the user is prompted for an instruction. If this was another file, command processing continues with the instruction following the CMDFILE command.

b) An EXIT command is processed. In this case, execution of BLOT terminates.

Note that command files can be nested; that is, a file that was called using a CMDFILE command can itself contain a CMDFILE command.

SHOW *command* <no default>

SHOW displays the plot parameter settings relevant to the *command.* For example, SHOW TMIN displays the time step selection parameter. This command must be issued within a subprogram (not at the BLOT level).

SHOW TIMES and SHOW STEPS display the selected time steps.

HELP *command* <no parameter>

HELP displays information about the program command given as the parameter. If no parameter is given, all the command verbs are displayed. HELP is system-dependent and may not be available on some systems.

5.1.2 Database Listing and Printing (GROPE) (any subprogram, including BLOT level)

SELECT *option* <no default>

SELECT selects the database information for the LIST and PRINT commands. These selections apply only to these two commands. Items are displayed in the order listed in the SELECT command. If the ADD parameter is present, the listed values are added to the current selection.

SELECT NODES [ADD,] $node\_range_1$, $node\_range_2$, ... <all nodes>

selects the nodes. Each *node_range* must be in the range form described at the beginning of this chapter.

SELECT ELEMENTS [ADD,] $element\_range_1$, $element\_range_2$, ...
<all elements>

selects the elements. Each *element_range* must be in the range form described at the beginning of this chapter. The selected element blocks are changed to select only those blocks that include a selected element. The selected elements are reordered so that they are grouped by element block.

SELECT BLOCKS or MATERIAL [ADD,] $block\_id_1$, $block\_id_2$, ...
<all element blocks>

selects the element blocks. The *block_id* is the element block identifier displayed by the LIST BLOCKS command. The selected elements are changed to select all the elements in the selected blocks and no elements in non-selected blocks.

SELECT NSETS [ADD,] set_id_1, set_id_2, ... <all node sets>

selects the node sets. The *set_id* is the node set identifier displayed by the LIST NSETS command.

SELECT SSETS [ADD,] set_id_1, set_id_2, ... <all side sets>

selects the side sets. The *set_id* is the side set identifier displayed by the LIST SSETS command.

SELECT HVARS [ADD,] $history\_variable_1$, $history\_variable_2$, ...
<all history variables>

selects the history variables.

SELECT GVARS [ADD,] $global\_variable_1$, $global\_variable_2$, ...
<all global variables>

selects the global variables.

SELECT NVARS [ADD,] *nodal_variable*$_1$, *nodal_variable*$_2$, ...
<all nodal variables>

selects the nodal variables.

SELECT EVARS [ADD,] *element_variable*$_1$, *element_variable*$_2$, ...
<all element variables>

selects the element variables.

SELECT READ *nstep* <1>

changes the selected time step to *nstep* steps forward (if *nstep* is positive) or backward (if *nstep* is negative) of the current selected step.

SELECT STEP *nstep* <current selected step>

changes the selected time step to step *nstep*.

SELECT TIME *time* <current selected step time>

changes the selected time step to the step that has the time nearest *time*.

LIST *option* <no default>

LIST displays the database information specified by *option* on the user's terminal. The "selected" items are specified with the SELECT command.

LIST VARS

displays a summary of the database. The summary includes the database title; the number of nodes, elements, and element blocks; the number of node sets and side sets; and the number of each type of variable.

LIST COORDINA

displays the coordinate values for the selected nodes.

LIST MAP

displays the element order map for all elements.

LIST BLOCKS or MATERIAL

displays a summary of the selected element blocks. The summary includes the block identifier, the number of elements in the block, the number of nodes per element, and the number of attributes per element.

LIST LINK or CONNECTI

displays the nodal connectivity of the selected elements.

LIST ATTRIBUT

displays the attribute values associated with the selected elements.

LIST NSETS

displays a summary of the selected node sets. The summary includes the set identifier and the number of nodes in the set.

LIST NNODES

displays the numbers of the nodes in the selected node sets.

LIST NFACTORS

displays the distribution factors of the nodes in the selected node sets.

LIST SSETS

displays a summary of the selected side sets. The summary includes the set identifier, the number of elements in the set, and the number of nodes in the set.

LIST SELEMS

displays the numbers of the elements in the selected side sets.

LIST SNODES

displays the numbers of the nodes in the selected side sets.

LIST SFACTORS

displays the distribution factors of the nodes in the selected side sets.

LIST QA

displays the QA records and the information records.

LIST NAMES

displays the names of the history, global, nodal, and element variables.

LIST HISTORY or HVARS

displays the values of the selected history variables for the selected time step.

LIST GLOBALS or GVARS

displays the values of the selected global variables for the selected time step.

LIST NODALS or NVARS

displays the values of the selected nodal variables for the selected nodes for the selected time step.

LIST ELEMENTS or EVARS

displays the values of the selected element variables for the selected elements for the selected time step.

LIST STEPS

displays the number of time steps and the minimum and maximum time step times.

LIST TIMES

displays the step numbers and times for all time steps on the database.

PRINT *option* <no parameter>

PRINT writes the database information specified by *option* to the print file. The valid *options* are described under the LIST command.

5.1.3 Time Step Selection (any subprogram)

The commands described in this section control the time step selection as explained in Section 2.5.

The following are the time step selection parameters:

- *tmin* is the minimum selected time,
- *tmax* is the maximum selected time,
- *nintv* is the number of selected time intervals, and
- *delt* is the selected time interval.

In the interval-times mode, up to *nintv* time steps at interval *delt* between *tmin* and *tmax* are selected. The mode may have a delta offset or a zero offset. With a delta offset, the first selected time is *tmin*+*delt*; with a zero offset, it is *tmin*.

In the interval-times mode with a delta offset, the number of selected time intervals *nintv* and the selected time interval *delt* are related mathematically by the equations:

$$delt = (tmax - tmin)/nintv \quad (1)$$
$$nintv = \mathsf{int}\,((tmin - tmax)/delt) \quad (2)$$

With a zero offset, *nintv* and *delt* are related mathematically by the equations:

$$delt = (tmax - tmin)/(nintv - 1) \quad (1)$$
$$nintv = \mathsf{int}\,((tmin - tmax)/delt) + 1 \quad (2)$$

The user specifies either *nintv* or *delt*. If *nintv* is specified, *delt* is calculated using equation 1. If *delt* is specified, *nintv* is calculated using equation 2.

In the all-available-times mode, all database time steps between *tmin* and *tmax* are selected (parameters *nintv* and *delt* are ignored). In the user-selected-times mode, the specified times are selected (all parameters are ignored).

The initial mode is the interval-times mode with a delta offset. Parameters *tmin*, *tmax*, and *nintv* are set to their default values and *delt* is calculated. The default values are defined in the command descriptions that follow.

TMIN *tmin* <minimum database time>

TMIN sets the minimum selected time *tmin* to the specified parameter value. If the user-selected-times mode is in effect, the mode is changed to the all-available-times mode.

In interval-times mode, if *nintv* is selected (by a NINTV or ZINTV command), *delt* is calculated. If *delt* is selected (by a DELTIME command), *nintv* is calculated.

TMAX *tmax* <maximum database time>

TMAX sets the maximum selected time *tmax* to the specified parameter value. If the user-selected-times mode is in effect, the mode is changed to the all-available-times mode.

In interval-times mode, if *nintv* is selected (by a NINTV or ZINTV command), *delt* is calculated. If *delt* is selected (by a DELTIME command), *nintv* is calculated.

NINTV *nintv* <10 or the number of database time steps − 1, whichever is smaller>

NINTV sets the number of selected time intervals *nintv* to the specified parameter value and changes the mode to the interval-times mode with a delta offset. The selected time interval *delt* is calculated.

ZINTV *nintv* <10 or the number of database time steps, whichever is smaller>

ZINTV sets the number of selected time intervals *nintv* to the specified parameter value and changes the mode to the interval-times mode with a zero offset. The selected time interval *delt* is calculated.

DELTIME *delt* $<(tmax - tmin)/(nintv - 1)$, where *nintv* is 10 or the number of database time steps, whichever is smaller>

DELTIME sets the selected time interval *delt* to the specified parameter value and changes the mode to the interval-times mode with a zero offset. The number of selected time intervals *nintv* is calculated.

ALLTIMES

ALLTIMES changes the mode to the all-available-times mode.

TIMES [ADD,] t_1, t_2, ... <no times selected>

TIMES changes the mode to the user-selected-times mode and selects times t_1, t_2, etc. The closest time step from the database is selected for each specified time.

Normally, a TIMES command selects only the listed time steps. If ADD is the first parameter, the listed steps are added to the current selected times. Any other time step selection command clears all TIMES selected times.

Up to the maximum number of time steps in the database may be specified. Times are selected in the order encountered on the database, regardless of the order the times are specified in the command. Duplicate references to a time step are ignored.

STEPS [ADD,] n_1, n_2, ... <no steps selected>

The STEPS command is equivalent to the TIMES command except that it selects time steps by the step number, not by the step time.

HISTORY ON or OFF <ON if allowed for current subprogram>

HISTORY controls whether history time steps are included in the selected time steps (if ON) or only whole time steps (if OFF). This command is only valid in subprograms which can process history variables (e.g., TPLOT).

For example, if the times from the database are 0.0, 0.5, 1.0, 1.5, etc., the commands

```
TMIN 0.0
TMAX 5.0
NINTV 5
```

select times 1.0, 2.0, 3.0, 4.0, and 5.0. If the NINTV command is replaced by

```
ZINTV 3
```

then times 0.0, 2.5, and 5.0 are selected. If the NINTV command is replaced by

```
DELTIME 2.0
```

then times 0.0, 2.0, 4.0 are selected.

5.1.4 Graphics Options (any subprogram)

QA ON or OFF <ON>, MESH or XY <current subprogram type>

QA controls whether the QA information is displayed on the plot legend (if ON) or not (if OFF). The QA information includes the database title and the creation and modification information. The line bordering the display is also included or omitted. This command controls the 2D plot legend if the second parameter is MESH or the curve plot legend if the parameter is XY. The XY parameter is only valid if issued from an X-Y curve subprogram.

AXIS ON or OFF <ON, initially OFF in 3D>, MESH or XY <current subprogram type>

AXIS controls whether the plot axes are numbered (if ON) or not (if OFF). This command controls the 2D plot axis if the second parameter is MESH or the curve plot axis if the parameter is XY. The XY parameter is only valid if issued from an X-Y curve subprogram.

LEGEND ON or OFF <ON>, MESH or XY <current subprogram type>

LEGEND controls whether the plot legend (excluding the QA information) is displayed on the plot (if ON) or not (if OFF). This command controls the 2D plot legend if the second parameter is MESH or the curve plot legend if the parameter is XY. The XY parameter is only valid if issued from an X-Y curve subprogram.

CAPTION *line_number* <0>, MESH or XY <current subprogram type>

CAPTION sets the three-line plot caption (up to 80 characters in each line) which is displayed at the bottom of the plot, with each line centered. The *line_number* is the number of the line to be changed. If *line_number* is zero, all three lines are changed. The caption line(s) must follow on the next line(s). If no caption is defined, the plot caption is blank.

This command controls the 2D plot caption if the second parameter is MESH or the curve plot caption if the parameter is XY. The XY parameter is only valid if issued from an X-Y curve subprogram.

The initial setting is no caption defined.

SOFTCHAR ON or OFF <ON>, *device* <1>

SOFTCHAR sets the plot character output mode to software characters (if ON) or hardware characters (if OFF). Hardware characters are drawn much faster,

but may produce inferior lettering on some devices. To change the character type for a secondary device, parameter *device* must be 2.

The initial setting is software characters for a hardcopy device or hardware characters for an interactive device.

This parameter is not changed by a RESET command or by switching subprograms.

FONT STICK or SANSERIF or ROMAN <STICK>, *device* <1>

FONT sets the font to be used for all software characters on a plot. The available fonts are stick, sanserif, and Roman. The stick font is much faster than the other two fonts. To change the font parameter for a secondary device, parameter *device* must be 2.

This parameter is not changed by a RESET command or by switching subprograms.

COLOR *ncol* <last selection>

COLOR sets the maximum number of standard color scale colors to use on a color plot to *ncol* (up to a maximum of 6 colors). The standard color scale colors are red, green, yellow, blue, magenta, and cyan.

The initial setting is the maximum number of colors available or six, whichever is smaller.

This parameter is not changed by a RESET command or by switching subprograms.

Issued from DETOUR:

> COLOR sets the number of contours to the number of colors if the spectrum color scale is undefined.

SPECTRUM *ncol* <5, initially 0>

SPECTRUM sets the maximum number of spectrum colors to use on a color plot to *ncol* (up to a maximum of 256-8 colors). The color table starting after color 8 is changed to a spectrum from blue to red. If *ncol* is zero, the spectrum color scale is undefined and the standard color scale is used.

This parameter is not changed by a RESET command or by switching subprograms.

Issued from DETOUR:

SPECTRUM sets the number of contours to the number of colors.

BACKGROUND *color* <no default, initially BLACK>

selects the background color for all plots. Valid colors are BLACK, WHITE, RED, GREEN, YELLOW, BLUE, CYAN, and MAGENTA. Unless explicitly done so by the user, the background color will not be assigned to any element blocks in 2D plots or assigned to any curves in X-Y plots.

FOREGROUND *color* <no default, initially WHITE>

selects the foreground color for all plots. Valid colors are BLACK, WHITE, RED, GREEN, YELLOW, BLUE, CYAN, and MAGENTA. The foreground color is used outline the plot frame, print the plot title and axis labels, display most of the information in the legend area, and outline the mesh and element blocks on 2D plots.

SNAP *nsnap* <1, initially 0>

SNAP sets the number of frames to be snapped on a camera device for each requested plot. The camera device must be "connected" in a system-dependent fashion before the program is run.

This parameter is not changed by a RESET command or by switching subprograms.

DISPVAR *option*

DISPVAR selects variables whose values will be displayed on the plot legend. Valid variables are the history and global variables, and the database time. There is not a limit to how many of these may be selected, but space may limit the number of values that actually appear on the legend. Initially, only the database time is displayed on the plot legend.

The DISPVAR command is valid only in the DETOUR and SPLOT subprograms. The RESET command, when issued from either of these subprograms, resets the list of display variables to include only the database time.

There are three forms of the DISPVAR command:

DISPVAR [ADD,] $variable_1$, $variable_2$, ... <no default>

selects the variables $variable_1$, $variable_2$, ... to have their values displayed on the plot legend. $variable_i$ may be any of the history or global variable names, or TIME (to specify the database time). If ADD is the first parameter, the listed variables are added to the current selected variables.

DISPVAR ALL <no default>

specifies that all allowable variables (all history and global variables plus the database time) will have their values displayed on the plot legend.

DISPVAR ON or OFF <toggles flag>

sets a flag specifying if the values of the display variables will be printed on the plot legend. This form of the command does not affect the list of display variables. Initially, this is set to ON.

5.1.5 Plot Set Display (any subprogram)

PLOT

PLOT generates the plots requested for the current plot set on the primary graphics device. No parameters are changed when the program returns to accept commands for the next plot set.

If the program is running interactively on a graphics terminal, the user is prompted for a response after each plot. The program expects one of the following responses:

- RETURN displays the next plot.
- "Q" (for quit) aborts the plot set and the program returns immediately to command input mode.
- "H" (for hardcopy) displays the plot shown on the graphics terminal on the hardcopy device (if any) and displays the next plot on the graphics terminal.
- "T" (for text) prompts for a location and a text string. The location is selected with the graphics cursor. The key pressed to set the position determines the text location. If the key is "C", the text string is centered over the selected location, otherwise the string is left-justified over the location. The text string is displayed on the graphics terminal and stored for output on the hardcopy device if a hardcopy plot is requested. The program again requests a response for the plot. Thus, the plot can be annotated with several text strings and then displayed on a hardcopy device.

When running interactively, the user can abort the current plot and return immediately to command input mode by issuing a system-defined interrupt (for example, CONTROL-C on the VAX).

If the program is running in batch mode or if the AUTO parameter is on, the program plots without user intervention.

HARDCOPY

HARDCOPY ends the command input for the current plot set and generates the plots requested on the hardcopy device. This command is appropriate only if two graphics devices are active. No parameters are changed when the program returns to accept commands for the next plot set.

AUTO ON or OFF <ON, initially OFF>

AUTO controls whether a response is requested from the user after each plot is completed on an interactive graphics terminal (if OFF) or whether all plots are displayed without user intervention (if ON). The responses expected are explained under the PLOT command.

This parameter is not changed by a RESET command or by switching subprograms.

5.2 Mesh Commands (any 2D or X-Y curve subprogram)

5.2.1 Mesh View Control (any 2D or X-Y curve subprogram)

The following commands set view-dependent parameters (see Section 3.2.1). They may be used in conjunction with the VIEW command.

The display mode commands (see Section 5.3.1) set these parameters.

EMPTY

EMPTY sets the selected view to empty, indicating that nothing is to be drawn for that view. This command is useful with multiple views.

The only way to change an empty view to a non-empty view is with a DEFORM command or a display mode command.

DEFORM ON or OFF <ON>

DEFORM sets whether the view displays the deformed mesh (if ON) or undeformed mesh (if OFF). The MAGNIFY command controls the deformation of all views.

Issued from DETOUR:

All display mode commands request the deformed mesh.

Issued from MESH and PATHLINE:

This command is not valid in these subprograms. The displacement magnification (set by the MAGNIFY command) is always zero and the mesh is always undeformed.

NUMBER NODES or ELEMENTS or ALL or OFF <ALL, initially OFF>

NUMBER sets the mesh numbering for the view. If the parameter is NODES or ALL, the node numbers are displayed on the mesh. If the parameter is ELEMENTS or ALL, the element numbers are displayed on the mesh. Only "active" elements (see Section 3.2.3) and nodes in at least one "active" element are numbered.

Issued from DETOUR:

All display mode commands turn off the numbering.

MLINES ON or DOTTED or OFF <ON>

MLINES controls whether all mesh lines (for elements in selected element blocks) are displayed (if ON or DOTTED) or only the mesh boundary and element block

boundaries (if OFF). If the parameter is DOTTED, the element boundary lines are dotted.

Issued from DETOUR:

All display mode commands set this parameter: the WIREFRAM and SOLID commands set it ON; all others set it OFF.

For example, the following command sequence:

```
XVIEW
YVIEW
VIEW 1 WIREFRAM
VIEW 2 WIREFRAM
VIEW 2 MLINES OFF
VIEW 3 CONTOUR SIGR
VIEW 3 MLINES DOTTED
VIEW 4 CONTOUR SIGR
PLOT
```

displays four views as follows:

- the mesh with all mesh lines,
- the mesh outline (the mesh boundary and element block boundaries),
- line contours on the mesh with dotted mesh lines, and
- line contours on the mesh outline.

BOUNDARY ON or BLACK or OFF <ON>

BOUNDARY controls whether the mesh boundary and element block boundaries are drawn in the foreground color (if ON) or black (if BLACK) or not drawn (if OFF). It is only valid if the mesh is to be painted. BOUNDARY OFF also sets the MLINES parameter to off; BOUNDARY ON sets the MLINES parameter to the default value for the current display mode.

Issued from DETOUR:

All display mode commands request the boundary be drawn in the foreground color. The BOUNDARY command is only valid if a solid mesh or painted contour display mode is active.

NSETS [ADD,] set_id_1, set_id_2, ... <all node sets>
NSETS OFF <initial condition>

NSETS selects the node sets to be displayed for the view. The side set selection is not affected. Only nodes in at least one "active" element (see Section 3.2.3)

are displayed.

The *set_id* is the node set identifier displayed by the LIST NSETS command.

If there is no parameter, all node sets are selected. If the parameter is OFF, no node sets are selected. If the first parameter is ADD, the sets listed are added to the current selected set. Otherwise, only the sets listed in the command are selected.

Issued from DETOUR:

All display mode commands set no node sets selected.

SSETS [ADD,] set_id_1, set_id_2, ... <all side sets>
SSETS OFF <initial condition>

The SSETS command is equivalent to the NSETS command except that it selects side sets instead of node sets. Only "active" elements (see Section 3.2.3) are displayed.

The LIST SSETS command displays the side set identifiers.

5.2.2 Active Element Control (any 2D or X-Y curve subprogram)

The following commands select active elements as described in Section 3.2.3.

Issued from DETOUR:

Only active elements are included in the calculation of the default contour parameters (see Section 5.3.2) for an element variable. When an active element control command is issued and contour mode of an element variable is selected, the contour parameters are recalculated unless a contour control command was issued after the CONTOUR or PAINT command. A CONTOUR or PAINT command with the variable as a parameter forces the recalculation of the contour parameters with the new active elements.

BLOCKS [ADD or DELETE,] *block_id*$_1$, *block_id*$_2$, ... <all element blocks>
BLOCKS OFF

BLOCKS selects the element blocks for active elements. Only elements in selected element blocks contribute to any variable plot (e.g., a contour plot). On a mesh plot, lines separating elements in the same element block (in some display modes) are only drawn for elements in a selected element block.

The *block_id* is the element block identifier displayed by the LIST BLOCKS command.

If there is no parameter, all element blocks are selected. If the parameter is OFF, no element blocks are selected. If the first parameter is ADD or DELETE, the element blocks listed are added to or deleted from the current selected set. Otherwise, only the element blocks listed in the command are selected.

MATERIAL [ADD or DELETE,] *block_id*$_1$, *block_id*$_2$, ... <all element blocks>
MATERIAL OFF

MATERIAL is exactly equivalent to a BLOCKS command.

VISIBLE [ADD or DELETE,] *block_id*$_1$, *block_id*$_2$, ... <all element blocks>

VISIBLE selects the element blocks that will be displayed in 3D. An element block that is not displayed is totally ignored in all plots and calculations.

The *block_id* is the element block identifier displayed by the LIST BLOCKS command.

If there is no parameter, all element blocks are visible. If the first parameter is ADD or DELETE, the element blocks listed are added to or deleted from the

current visible set. Otherwise, only the element blocks listed in the command are visible.

DEATH ON <no default>, *state_variable* <last selection>
DEATH OFF <initial condition>

DEATH turns on or off the capability to ignore "dead" elements.

DEATH ON

allows elements to die and be born at the start of a time step. A "dead" element is ignored in all calculations and plots, including a simple mesh plot. "Dead" nodes can be displayed with the DEADNODE command. The *state_variable* is an element database variable that gives the state of each element for each time step (zero if alive, otherwise dead).

The initial default *state_variable* name is DEATH.

DEATH OFF

causes all elements to be alive for all time steps.

Issued from MESH and PATHLINE:

This command is not valid in these subprograms.

DEADNODE ON or OFF <ON>

DEADNODE determines whether "dead" nodes are to be displayed on all mesh plots (if ON) or not (if OFF). A dead node is a node which is not connected to any element which is alive. This parameter is active only if element birth/death is active (see the DEATH command). This command does not affect the contour calculation.

Issued from MESH and PATHLINE:

This command is not valid in these subprograms.

5.2.3 Multiple Views Control (any 2D or X-Y curve subprogram)

The following commands control multiple views on one plot as explained in Section 3.2.1. The initial setting is one view. Symmetric views are not allowed in 3D.

VIEW *nview* <no default>, *view_dependent_command* <no default>

The VIEW command is used when multiple views have been created using the XVIEW or YVIEW commands. Each of the views is assigned an integer identification from 1 to 4 according to Figure 3.1. Some graphics parameters can vary between views (the description for a command will specify if it is for a view-dependent parameter). The VIEW command must be used to set a view-dependent parameter when more than one view is defined. *view_dependent_command* specifies the name of the command associated with the view-dependent parameter to be set, and *nview* specifies the identifier of the view for which the parameter is to be set. If *nview* is zero, the parameter is set for all defined views.

XVIEW ON or OFF <ON, initially OFF>

XVIEW defines (if ON) or deletes (if OFF) non-symmetric vertically divided views. If the parameter is ON, new views are created to the left of the existing views, and the new views have the view-dependent parameters of the existing views beside them. If the parameter is OFF, the left views are deleted.

YVIEW ON or OFF <ON, initially OFF>

YVIEW defines (if ON) or deletes (if OFF) non-symmetric horizontally divided views. If the parameter is ON, the new views are created below the existing views, and the new views have the view-dependent parameters of the existing views above them. If the parameter is OFF, the lower views are deleted.

XSYM LEFT or RIGHT <last selection>, *xsym* <last selection>
XSYM OFF <initial condition>

XSYM defines or deletes symmetric vertically divided views.

XSYM LEFT or RIGHT

defines the symmetric vertically divided views. The LEFT/RIGHT parameter determines if the new views are created to the left or right of the existing views. The existing views are reflected around the vertical symmetry axis along the horizontal coordinate *xsym*. The new views have the view-dependent parameters of the existing views beside them (unless vertically divided views are already defined).

If the LEFT/RIGHT parameter changes, *xsym* defaults to the minimum horizontal mesh coordinate (if LEFT) or to the maximum horizontal mesh coordinate (if RIGHT). Otherwise *xsym* defaults to its last selected value. The initial default is LEFT with *xsym* at the minimum horizontal coordinate value.

XSYM OFF

deletes the symmetric vertically divided views defined by a previous XSYM command. The left views are deleted if the views were defined with XSYM LEFT; otherwise the right views are deleted (and the left views are renumbered).

YSYM BOTTOM or TOP <last selection>, *ysym* <last selection>
YSYM OFF <initial condition>

YSYM defines or deletes symmetric horizontally divided views.

YSYM BOTTOM or TOP

defines the symmetric horizontally divided views. The BOTTOM/TOP parameter determines if the new views are created above or below the existing views. The existing views are reflected around the horizontal symmetry axis along the vertical coordinate *ysym*. The new views have the view-dependent parameters of the existing views above or below them (unless horizontally divided views are already defined).

If the BOTTOM/TOP parameter changes, *ysym* defaults to the minimum vertical mesh coordinate (if BOTTOM) or to the maximum vertical mesh coordinate (if TOP). Otherwise *ysym* defaults to its last selected value. The initial default is BOTTOM with *ysym* at the minimum vertical coordinate value.

YSYM OFF

deletes the symmetric horizontally divided views defined by a previous YSYM command. The lower views are deleted if the views were defined with YSYM BOTTOM; otherwise the upper views are deleted (and the lower views are renumbered).

MULTTIME ON or OFF <ON, initially OFF>

MULTTIME ON assigns a different time step to each non-empty view. Thus, if four views are defined, the first four selected times appear on the first plot (in the order that the views are ordered), the next four times are on the second plot, etc.

MULTTIME OFF assigns a single time step to all defined views.

Issued from MESH and PATHLINE:

This command is not valid in these subprograms.

5.2.4 Mesh Control (any 2D or X-Y curve subprogram)

MAGNIFY *dfac* <1.0>

MAGNIFY causes the displacements to be magnified by *dfac* before they are added to the coordinates. If the parameter is zero, the mesh is not deformed.

Issued from MESH and PATHLINE:

This command is invalid in these subprograms. The magnification is zero and may not be set.

When switching from a subprogram which cannot display the deformed mesh to one that can, the magnification is set to the last non-zero value for this parameter.

HIDDEN *level* <3>

HIDDEN controls the 3D hidden line and surface algorithm. The hidden *level* is defined below.

- 0 indicates that hidden lines are not removed. This option is not valid with painted surfaces.
- 1 requests a simple hidden surface algorithm that removes any surface (and its lines) whose normal faces away from the viewer.
- 2 requests the normal hidden node algorithm, and removes any surface (and its lines) if any node of the surface is hidden.
- 3 requests the normal hidden node algorithm and displays partial lines. A surface is considered hidden if any of its nodes are hidden.
- 4 requests the algorithm used for level 3. On painted plots, all surfaces with some hidden nodes are ordered by the Z coordinates and processed as visible surfaces.

ZOOM *hmin*, *hmax*, *vmin*, *vmax* <deformed mesh limits + 5% of range> or CURSOR
ZOOM TRANSLAT *hcen*, *vcen* <unchanged> or CURSOR
ZOOM EACH <default for 3D>
ZOOM MESH <default for 2D>

ZOOM sets the mesh window limits.

ZOOM *hmin*, *hmax*, *vmin*, *vmax* or CURSOR

sets the horizontal mesh window limits to *hmin* and *hmax* and the vertical limits to *vmin* and *vmax*. If the first parameter is a number, the mesh

limits are entered as parameters. If the first parameter is CURSOR, the mesh limits are selected with the graphics cursor. The left, bottom position (*hmin*, *vmin*) is selected first, then the right, top position (*hmax*, *vmax*).

ZOOM TRANSLAT *hcen*, *vcen* or CURSOR

moves the current window limits such that the window is centered on *hcen*, *vcen*. If the second parameter is CURSOR, the center is selected with the graphics cursor.

ZOOM EACH

causes the window limits to be reset for each plot to the smallest (expanded) window that encloses the displayed mesh.

ZOOM MESH

sets the window limits to the (expanded) deformed mesh limits or the undeformed mesh limits if all views are undeformed.

If the SQUARE parameter is on, the mesh window defaults to a square or two rectangles forming a square (see Section 3.2.1). To accomplish this, the shorter of the horizontal or vertical range is expanded with the specified range centered on the window. If the SQUARE parameter is off, the specified limits exactly define a rectangular window.

SQUARE ON or OFF <ON>

SQUARE determines if the mesh window will be square or rectangular. If the parameter is ON, the window limits set by the ZOOM command are expanded to form a square or two rectangles forming a square (see Section 3.2.1). If the parameter is OFF, the limits of the mesh window are the limits set by the ZOOM command (forming a rectangular window).

TICK *tick* <0.0>

TICK sets the axis tick interval for both mesh axes to *tick*. If *tick* is zero, the tick interval is automatically calculated from the mesh window limits.

Automatic calculation is selected when a ZOOM command is issued.

BLKCOL or **MATCOL** *option*

BLKCOL assigns colors to element blocks in wireframe and solid mesh plots. Valid colors are BLACK, WHITE, RED, GREEN, YELLOW, BLUE, CYAN, and MAGENTA. A color need not be assigned to all element blocks. If an element block does not have a color assigned to it, BLOT will choose a color

(other than BLACK or WHITE) from those that have not been assigned to an element block by the user. If the device being plotted to does not support a particular color, BLOT treats the corresponding element block as if the user had not assigned it a color.

The RESET command has no effect on the element block selections made.

There are three forms of the BLKCOL command:

BLKCOL or MATCOL $color\_spec_1$, $color\_spec_2$, ... <no default>

$color\_spec_i$ has the form

$element\_block\_range_1$, $element\_block\_range_2$, ... $color\_name$

and assigns the color *color_name* to the element blocks specified by $element\_block\_range_1$, $element\_block\_range_2$, ...

BLKCOL or MATCOL RESET <no default>

deletes all user assigned colors. Default colors are assigned by BLOT.

BLKCOL or MATCOL ON or OFF <toggles flag>

sets a flag specifying if the user assigned colors should be used in plotting. This form of the command does not affect the user defined colors. Initially, this is set to OFF. It is set to ON by the first form of the BLKCOL command and to OFF by the second.

LINETHICKNESS $spec_1$, $spec_2$, ...

LINETHICKNESS assigns thicknesses to the three types of mesh lines that appear on mesh plots: mesh boundary lines, element block boundary lines, and element boundary lines.

$spec_i$ has the form *line_type*, *line_thickness* and assigns the line thickness specified by *line_thickness* to the lines of type specified by *line_type*. *line_type* may be

- MESH to specify mesh boundary lines.
- BLOCK to specify block boundary lines.
- ELEMENT to specify element boundary lines.
- ALL to specify all boundary lines.

line_thickness may be

- THIN to specify thin lines.
- MEDIUM to specify lines with medium thickness.
- THICK to specify thick lines.

- *value* to specify lines with thickness *value*. *value* is a real number in the range 0.0 to 1000.0, where line thickness increases with increasing value.

SPHERE ON or OFF <toggles flag>

SPHERE sets a flag specifying if all elements in the database are to be plotted as outlined circles. If the flag is on, then all elements in the database must have only one node that defines its center and the first attribute of the element must be its radius.

FSPHERE ON or OFF <toggles flag>

FSPHERE sets a flag specifying if all elements in the database are to be plotted as filled circles. If the flag is on, then all elements in the database must have only one node that defines its center and the first attribute of the element must be its radius.

5.2.5 3D Rotation Commands (any 2D or X-Y curve subprogram)

ROTATE $axis_1$, $ndeg_1$, $axis_2$, $ndeg_2$, ... <last selection>
ROTATE RESET <initial condition>

ROTATE rotates the 3D mesh. Each (*axis*, *ndeg*) parameter pair specifies an axis (X or Y or Z) and the number of degrees to rotate. The axis refers to the screen axis, not to the object axis. The rotations are according to right-hand rule.

Rotations are cumulative. The ROTATE RESET command resets to no rotation. The EYE command also sets the mesh rotation.

The rotation is not changed by switching subprograms, but it is reset by a RESET command.

EYE *xpos*, *ypos*, *zpos* <last selection>

EYE rotates the 3D mesh so that the viewer's "eye" position is at (*xpos*,*ypos*,*zpos*) (in object coordinates). The ROTATE command also sets the mesh rotation. The current EYE position is displayed after each ROTATE command.

The rotation is not changed by switching subprograms, but it is reset by a RESET command.

CENTER *xcen*, *ycen*, *zcen* <last selection>

CENTER moves the 3D center of rotation to (*xcen*,*ycen*,*zcen*) (in object coordinates).

The center of rotation is not changed by switching subprograms, but it is reset by a RESET command.

5.3 DETOUR Commands (DETOUR subprogram)

5.3.1 Mesh Display Mode Control (DETOUR subprogram)

The following commands set the display mode as explained in Section 3.4.2. The display mode is a view-dependent parameter (see Section 3.2.1). The display mode commands may be used in conjunction with the VIEW command. One and only one display mode is active for each view. Thus, the display mode commands supersede one another. The initial display mode is wireframe mesh mode.

Modes which reference a variable use only values for "active" elements (see Section 3.2.3). The specified variable applies to all views (see Section 3.2.1).

Display mode commands affect the mesh view control parameters (see Section 5.2.1).

WIREFRAM

WIREFRAM sets the view to wireframe mesh mode, which displays the mesh lines.

SOLID

SOLID sets the view to solid mesh mode, which paints each element, using a different color for each element block. All visible element blocks are painted, even if the block is not selected.

CONTOUR *variable* <last selected variable>

CONTOUR sets the view to line contour mode, which plots contours of the *variable* on the mesh.

The *variable* may be either a nodal or element database variable. Contours are plotted for nodal values only. An element variable is converted to a nodal variable by averaging the values for all active elements containing the node.

If a variable is specified, the contour parameters are recalculated as explained in Section 5.3.2. The contour parameters are also recalculated if the current parameters were calculated with a PAINT command, unless a contour control command was issued after the PAINT command.

PAINT *variable* <last selected variable>

PAINT sets the view to paint contour mode, which paints contours of the *variable* on the mesh.

The *variable* may be either a nodal or element database variable. Contours are plotted for nodal values only. An element variable is converted to a nodal variable by averaging the values for all active elements containing the node.

If a variable is specified, the contour parameters are recalculated as explained in Section 5.3.2. The contour parameters are also recalculated if the current parameters were calculated with a CONTOUR command, unless a contour control command was issued after the CONTOUR command.

VECTOR *x_variable* or 0, *y_variable* or 0, *z_variable* or 0 (if 3D) <last selected variables>

VECTOR sets the view to vector mode, which plots a vector representing the components *x_variable*, *y_variable*, and *z_variable* (for a 3D mesh) on the mesh. The variables may be either nodal or element database variables, but all must be of the same type. If the variables are nodal variables, a vector is plotted at each node. If the variables are element variables, a vector is plotted at the center of each active element. If a parameter is 0, the corresponding vector component is zero (a scalar plot).

The vector lengths are controlled by the vector/symbol scale factor (see the VSCALE command). This factor is set to 1.0 by the VECTOR command.

SYMBOL CRACK or ANGLE <no default>, *variable* <last selected variable>

SYMBOL sets the view to element symbol mode. This special purpose option allows the user to plot a symbol on the mesh at the center of each active element.

The *variable* is an element database variable which gives a value for each element. If the first parameter is CRACK, DETOUR plots an "X" for a negative value and an "O" for a non-zero positive value (nothing is plotted for a zero value). If the first parameter is ANGLE, DETOUR plots a line at the indicated angle (in degrees) if the angle is between -360 and 360 and an "*" if the angle is less than -360 (nothing is plotted if the angle is greater than 360). An angle of zero plots a horizontal line; positive angles are measured counterclockwise from the horizontal.

The symbol size may be controlled the vector/symbol scale factor (see the VSCALE command). This factor is set to 1.0 by the SYMBOL command.

variable

A database variable name is equivalent to a display mode command if the variable name does not conflict with any command or command abbreviation. If the

current display mode is wireframe or solid mesh mode, this command is equivalent to a CONTOUR command. If the display mode references a variable (e.g., element symbol mode), this command is equivalent to the command which sets the current display mode. For example, if the view is in vector mode, the command

SIGR, SIGZ

is equivalent to the command

VECTOR SIGR, SIGZ.

5.3.2 Contour Control (DETOUR subprogram)

The commands in this section control a contour plot. The following are the contour parameters:

- *ncntr* is the number of contours,
- *delc* is the contour interval,
- *cmin* is the minimum contour value, and
- *cmax* is the maximum contour value.

DETOUR plots *ncntr* contours at uniform intervals *delc* from *cmin* through *cmax*. The *ncntr* parameter has the highest priority, meaning that all other parameters change with respect to it. The only way to change the number of contours is with the NCNTRS command or the COLOR and SPECTRUM commands.

In line contour mode, the contour parameters are related mathematically by the equation:

$$delc = (cmax - cmin)/(ncntr - 1)$$

In paint contour mode, the contour parameters are related mathematically by the equation:

$$delc = (cmax - cmin)/ncntr$$

The contour parameters may be reset to default values when the contour variable changes or when the active element criteria changes. The number of contours *ncntr* is not changed when the default is calculated. The minimum and maximum database variable values are used to compute a rounded default contour interval *delc*. In line contour mode, the default contour limits *cmin* and *cmax* are approximately one-half *delc* inward from the database variable limits. In paint contour mode, the default contour limits *cmin* and *cmax* are slightly outside the database variable limits.

If the contour mode changes from line to painted contour mode or from painted to line contour mode and the default is not calculated, the contour maximum *cmax* is recalculated, but the rest of the parameters remain unchanged.

NCNTRS *ncntr* <6>

NCNTRS sets the number of contours *ncntr* to the specified parameter value. A new contour interval *delc* is calculated from the variable limits, but it is not rounded.

CRANGE *cmin* <last selection>, *cmax* <last selection>

CRANGE sets the minimum contour value *cmin* and the maximum contour value *cmax* to the specified parameter values. The contour interval *delc* is calculated.

DELCNTR *delc* <last selection>, *cmin* <last selection>

DELCNTR sets the contour interval *delc* and the contour minimum *cmin* to the specified parameter values. The contour maximum *cmax* is calculated.

CSHIFT *cval* <*cmin*>

CSHIFT changes the contour limits *cmin* and *cmax* so that the parameter *cval* falls exactly on the nearest contour line or contour paint boundary. The contour interval *delc* is not affected.

5.3.3 Mode Specific Options (DETOUR subprogram)

CLABEL ON or *label_incr* or OFF <ON>

CLABEL controls the density of identifying letters on a contour plot. This parameter is active only in line contour mode. If the parameter is OFF, no contour letters appear on the plot. If the parameter is ON or a number, contour letters appear where a contour line intersects the mesh boundary or an element block boundary. If the parameter is a number (*label_incr*), letters appear where a contour line intersects an element boundary, with the density of these letters decreasing as *label_incr* increases. For example, if *label_incr* is 1, a letter appears at every intersection with an element boundary; if *label_incr* is 4, a letter appears at every 4^{th} intersection.

COPEN ON or OFF <ON, initially OFF>, ON or OFF <ON, initially OFF>

COPEN controls the plotting of values outside the contour limits on a painted contour plot. The first parameter controls whether values below the minimum parameter value should be painted in the minimum color (if ON) or left unpainted (if OFF). The second parameter controls whether values above the maximum parameter value should be painted in the maximum color (if ON) or left unpainted (if OFF).

CSYMBOLS *nmin* <10>, *nmax* <*nmin*>

CSYMBOLS controls the plotting of symbols at the minimum and maximum nodal variable values on a contour plot and the display of these values on the plot legend. Parameters *nmin* and *nmax* are active only in the contour modes. Symbols are displayed at the minimum (with the values in the legend) only if the number of minimums for the plot is less than or equal to *nmin*. The maximum symbols and values are displayed only if the number of maximums is less than or equal to *nmax*. The number of minimums and maximums includes the entire mesh, not just the mesh visible in the window. Setting *nmin* or *nmax* to zero disables the minimum or maximum display.

VSCALE *scale_factor* <1.0>

VSCALE sets the vector/symbol scale factor. This parameter is active only in vector or element symbol mode.

In vector mode, the magnitudes of the vectors are multiplied by *scale_factor* and by the default vector scaling factor to determine their length. The default scaling factor is five percent of the window coordinate range divided by the maximum

absolute value of all the vector component variables. Thus, the vectors maintain the same length in relation to the window size.

In element symbol mode, *scale_factor* is multiplied by the constant symbol scaling factor to determine the symbol size.

The VECTOR and SYMBOL commands reset this parameter to 1.0.

5.4 PATHLINE Commands (PATHLINE subprogram)

ADD

ADD saves the pathlines defined in the last plot set and causes future definitions to be added to the set. This command is ignored if new pathlines have been defined for the plot set.

REMOVE *pathline*$_1$, *pathline*$_2$, ... <no default>

REMOVE deletes the specified pathlines. Each parameter *pathline*$_i$ is the number of a pathline that is to be deleted. The pathline number is displayed with the LIST PLOT command. If *pathline*$_i$ is negative, the last -*pathline*$_i$ pathlines defined are deleted.

LOCATION *x_variable*, *y_variable*, *z_variable* (if 3D), <no default>*range*$_1$, *range*$_2$, ... <no default>

LOCATION defines a pathline. The pathline is defined by the variables x_variable and y_variable (and z_variable if a 3D mesh). Each time step in the database defines a point on the pathline, and x_variable gives the x coordinate, y_variable gives the y coordinate, and z_variable gives the z coordinate. The variables may be history, global or nodal variables, but all must be of the same type. If the variables are nodal variables, a set of nodes must also be selected. Each *range* must be in the range form described at the beginning of this chapter.

The first LOCATION command issued for a plot set starts a new set of pathlines (unless an ADD command is entered first). All following LOCATION commands are cumulative.

5.5 X-Y Curve Commands (any X-Y curve subprogram)

5.5.1 Curve Definition (any X-Y curve subprogram)

ADD

ADD saves the curves defined in the last plot set and causes future definitions to be added to the set. This command is ignored if new curves have been defined for the plot set.

REMOVE $curve_1$, $curve_2$, ...<no default>

REMOVE deletes the specified curves. Each parameter $curve_i$ is the number of a curve that is to be deleted. The curve number is displayed with the LIST PLOT command. If $curve_i$ is negative, the last $-curve_i$ curves defined are deleted.

5.5.2 Axis Labeling and Scaling (any X-Y curve subprogram)

The axis label, scaling, and tick interval may be reset to their default values when the axis changes.

Issued from TPLOT:

The X axis changes when a new X variable is requested. The Y axis changes when a new Y variable is requested. New variables are requested with a TYPLOT or XYPLOT command.

Issued from SPLOT:

The X axis changes when a NODES or ELEMENTS command is issued. The Y axis changes when an SYPLOT command is issued.

XLABEL

XLABEL sets the label (up to 40 characters) to be centered below the X axis. The label must follow on the next line. If no label is defined, a default is provided.

Issued from TPLOT:

For time history plots, the default X axis label is "TIME". For variable-versus-variable plots, if all curves on the plot reference the same X variable, the default is the variable name; otherwise no label is provided.

Issued from SPLOT:

The default X axis label is "DISTANCE".

The default label is selected when the X axis changes if no XLABEL command has been issued for this plot set.

YLABEL

YLABEL sets the label (up to 40 characters) to be centered to the left of the Y axis. The label must follow on the next line. If no label is defined, a default is provided.

Issued from TPLOT:

If all curves on the plot reference the same Y variable, the default Y axis label is the variable name; otherwise no label is provided.

Issued from SPLOT:

If there is only one curve on the plot, the default Y axis label is the curve variable name; otherwise no label is provided.

The default label is selected when the Y axis changes if no YLABEL command has been issued for this plot set.

XSCALE *xmin* <last selection>, *xmax* <last selection>

XSCALE sets the X axis minimum and maximum values to *xmin* and *xmax*. If neither value is given, the axis is automatically scaled from the data.

Automatic scaling is selected when the X axis changes if no XSCALE command has been issued for this plot set.

YSCALE *ymin* <last selection>, *ymax* <last selection>

YSCALE sets the Y axis minimum and maximum values to *ymin* and *ymax*. If neither value is given, the axis is automatically scaled from the data.

Automatic scaling is selected when the Y axis changes if no YSCALE command has been issued for this plot set.

XTICK *xtick* <0.0>

XTICK sets the X axis tick interval to *xtick*. If *xtick* is zero, the tick interval is automatically calculated from the data.

Automatic calculation is selected when the X axis changes if no XTICK command has been issued for this plot set.

YTICK *ytick* <0.0>

YTICK sets the Y axis tick interval to *ytick*. If *ytick* is zero, the tick interval is automatically calculated from the data.

Automatic calculation is selected when the Y axis changes if no YTICK command has been issued for this plot set.

SAMESCAL ON or OFF <ON, initially OFF>

SAMESCAL controls whether individual curves in the plot set are to have the same scale (if ON) or be scaled individually (if OFF). This parameter has no meaning if the axis scales have been set with the XSCALE and YSCALE commands. A SAMESCAL command overrides the NORMAL parameter.

Issued from TPLOT:

If the curves are overlaid, they all have the same scale.

Issued from SPLOT:

Since the distances are constant over the entire plot set, the X axis always has the same scale. If the parameter is ON, the curves for all variables and all times have the same scale.

If the parameter is OFF and the curves are overlaid, all the curves on one plot have the same scale, but each variable (if times are overlaid) or each time step (if variables are overlaid) is scaled individually.

NORMAL ON or OFF <ON, initially OFF>

NORMAL controls whether individual curves on a plot are to have the same scale (if OFF) or be scaled individually (if ON). If the parameter is OFF, the scale for each curve is normalized to the same scale (0..1). This parameter has no meaning if the axis scales have been set with the XSCALE and YSCALE commands. A NORMAL command overrides the SAMESCAL parameter.

Issued from SPLOT:

Since the distances are constant over the entire plot set, the X axis always has the same scale.

RATIOXY *ratio* <1.0>

RATIOXY sets the X-to-Y axis length ratio to *ratio*. For example, if *ratio* is 0.5, the X axis is half as long as the Y axis.

5.5.3 Curve Display Options (any X-Y curve subprogram)

OVERLAY VARIABLE (ON) or TIME or OFF <VARIABLE, initially OFF>

OVERLAY controls whether curves are overlaid on one plot. If the parameter is VARIABLE, the curves for all variables are displayed on one plot for each time step. If the parameter is TIME, the curves for all time steps are displayed on one plot for each requested variable. If the parameter is OFF, each curve is displayed on a separate plot.

Issued from TPLOT:

The TIME option is not valid. If the parameter is VARIABLE, all curves are displayed on one plot.

Issued from SPLOT:

If the parameter is VARIABLE, the curves for all variables are displayed on one plot for each time step. If the parameter is TIME, the curves for all time steps are displayed on one plot for each requested variable.

GRID ON or OFF <ON, initially OFF>

GRID controls whether a grid is drawn on the plot (if ON) or not (if OFF). The placement of the grid lines may be controlled with the XTICK and YTICK commands.

LINES ON or VARY or OFF <ON>

LINES controls the type of line plotted for each curve. If the parameter is ON, a solid line is plotted. If the parameter is VARY, the line type changes for each curve on a plot. The line type cycles through six types of lines: solid, dotted, dot dash, short dash, long dash, and medium dash. If the parameter is OFF, only symbols are displayed (see the SYMBOLS command).

This parameter is set ON if it is OFF and a SYMBOLS OFF command is issued.

SYMBOLS VARY or *symbol_number* or OFF <VARY, initially OFF>

SYMBOLS controls whether symbols are plotted at each data point on a curve. If the parameter is VARY (or ON), symbols are plotted, with the symbol changing for each curve on a plot. The symbol cycles through the six symbols listed below. If the parameter is OFF, only lines are plotted (see the LINES command). If the parameter is a number, the symbol corresponding to *symbol_number* is plotted for all curves.

The symbols with their identifying numbers are:

1) square
2) diamond
3) cross
4) x
5) triangle
6) circle

This parameter is set to VARY if it is OFF and a LINES OFF command is issued.

CRVNUM FIRST or MIDDLE or LAST or OFF <LAST>

CRVNUM controls where the curves are numbered. If the parameter is FIRST, the number appears to the left of the first point of each curve. If the parameter is MIDDLE, the number appears over the middle point of each curve. If the parameter is LAST, the number appears to the right of the last point of each curve. If the parameter is OFF, no curve numbers appear.

Curves are only numbered if the number of line types or symbols selected is less than the number of curves on the plot, regardless of the setting of this parameter.

5.5.4 Neutral File Options (any X-Y curve subprogram)

The neutral file format is the format specified for input to the GRAFAID plot program [9]. Each defined curve is written to the neutral file as a set of X-Y pairs. Some descriptive information is also written with each curve. Many commands (e.g., XSCALE and XLABEL) affect the output of this descriptive information. Appendix B describes the format in detail.

A name for each curve is written to the neutral file. This name can be used in GRAFAID to reference the curve. The neutral file curve name consists of a character string with a number appended. The user may set the string and the number.

ACURVE *acurve* <blank>

ACURVE sets the name part of the neutral file curve name to *acurve* (up to 8 characters). The neutral file curve name is the curve name with the curve number (set by the NCURVE command) appended. This name is written to the neutral file.

This parameter is not changed by a RESET command or by switching subprograms.

NCURVE *ncurve* <1>, *inccrv* <1>

NCURVE sets the starting neutral file curve number to *ncurve* and the increment for this number to *inccrv*. The neutral file curve name is the curve name (set by the ACURVE command) with the curve number appended. This name is written to the neutral file. The curve number is incremented after each curve is written to the neutral file.

This parameter is not changed by a RESET command or by switching subprograms.

NEUTRAL

NEUTRAL ends the command input for the current plot set and writes the defined curves to the neutral file. No parameters are changed when the program returns to accept commands for the next plot set.

For example, the TPLOT commands

```
SIGX 50 TO 250 BY 50
ACURVE SIGX
NCURVE 50,50
NEUTRAL
```

assign the name SIGX50 to the first curve, SIGX100 to the second curve, etc.

5.5.5 Mesh Display (any X-Y curve subprogram)

MESH

MESH displays the mesh from an X-Y curve subprogram using the plot parameter settings defined for the MESH subprogram. The user may be prompted for a response after the mesh is displayed. The responses expected are explained under the PLOT command.

5.6 TPLOT Commands (TPLOT subprogram)

5.6.1 Curve Definition (TPLOT subprogram)

TYPLOT

TYPLOT defines a time history curve. The variable with its list of node or element ranges (if the variable is a nodal or element variable) must follow on the next line. The times are specified by the time step selection commands. The first TYPLOT or XYPLOT command issued for a plot set starts a new set of curves (unless an ADD command is entered first). All following TYPLOT and XYPLOT commands are cumulative.

Each node or element range must be in the range form described at the beginning of this chapter. One TYPLOT command may define several curves of different nodes or elements of the same variable. For example, the command

```
TYPLOT
SIGR 56, 40 TO 20 BY -10, 5
```

defines five curves of variable SIGR for elements 56, 40, 30, 20, and 5, in that order.

TYPLOT may reset the XLABEL, YLABEL, XSCALE, YSCALE, XTICK, and YTICK parameters to their default values.

variable node_range$_1$ or *element_range*$_1$, *node_range*$_2$ or *element_range*$_2$, ... <no number>

A database variable name with its list of node or element ranges (if the variable is a nodal or element variable) is equivalent to a TYPLOT command if the variable name does not conflict with any command or command abbreviation. For example, the command

```
SIGMAX, 1, 5
```

is equivalent to the command

```
TYPLOT
SIGMAX, 1, 5.
```

XYPLOT

XYPLOT defines a variable-versus-variable curve. The X variable with its list of node or element ranges (if the variable is a nodal or element variable) must follow

on the next line. The Y variable with its list of node or element ranges (if the variable is a nodal or element variable) must be on the following line. The first TYPLOT or XYPLOT command issued for a plot set starts a new set of curves (unless an ADD command is entered first). All following TYPLOT and XYPLOT commands are cumulative.

Each X and Y node or element range must be in the range form described at the beginning of this chapter. One XYPLOT command may define several curves of different nodes or elements of one X and one Y variable. If multiple nodes or elements are requested on both the X and Y variables, the number of nodes or elements selected for the X and Y variables must be equal. For example, the command

```
XYPLOT
SIGMAX, 1
SIGZ, 10 TO 20 BY 5, 30
```

defines four curves of SIGMAX at element 1 along the X axis versus SIGZ at elements 10, 15, 20, and 30 along the Y axis.

XYPLOT may reset the XLABEL, YLABEL, XSCALE, YSCALE, XTICK, and YTICK parameters to their default values.

5.7 SPLOT Commands (SPLOT subprogram)

5.7.1 Curve Definition (SPLOT subprogram)

NODES [ADD,] [PATH,] *node_range*$_1$, *node_range*$_2$, ... <no default>

NODES selects the nodes which define the plotting distances and the variable values to plot. A NODES or ELEMENTS command must be issued before an SYPLOT command.

Each *node_range* must be in the range form described at the beginning of this chapter. For example, NODES 40 TO 20 BY -10, 7 selects nodes 40, 30, 20, and 7, in that order.

If ADD is the first parameter, the listed nodes are added to the current selection. A NODES command with the ADD parameter is preferable to continuing a NODES command over two lines.

If the parameter PATH proceeds any node number, nodes are selected which form a "path" between each pair of specified nodes. The next node along the path is the node connected to the current path node that is closest to the line connecting the path endpoints and closer to the endpoint than the current node. Connected nodes are in the same element (including nodes which are diagonal from each other).

The existing defined curves are deleted if and only if the selected curve variables are not nodal variables.

NODES may reset the XLABEL, XSCALE, and XTICK parameters to their default values.

ELEMENTS [ADD,] [PATH,] *element_range*$_1$, *element_range*$_2$, ... <no default>

The ELEMENTS command is equivalent to the NODES command except that it selects elements instead of nodes. Element paths may be defined. Connected elements have at least one node in common.

SYPLOT *variable* <no default>

SYPLOT defines a distance-versus-variable curve. A NODES or ELEMENTS command must be issued before this command. The *variable* must be a nodal database variable if nodes are selected or an element database variable if elements are selected. The first SYPLOT command issued for a plot set starts a new set of curves (unless an ADD command is entered first). All following SYPLOT commands are cumulative.

SYPLOT may reset the YLABEL, YSCALE, and YTICK parameters to their default values.

variable

A database variable name is equivalent to a SYPLOT command if the variable name does not conflict with any command or command abbreviation. For example, the command

SIGMAX

is equivalent to the command

SYPLOT SIGMAX.

5.7.2 Mesh Display (SPLOT subprogram)

ECHO

ECHO displays the mesh showing the selected nodes or elements. A NODES or ELEMENTS command must be issued before this command. The mesh window is scaled to form a box around the selected nodes or elements. The selected nodes or elements are numbered and connected by arrows.

6. Informational and Error Messages

Error messages may appear during the program initialization, during command input or during the plotting. If a fatal error occurs, the program aborts.

An error message of the following form may appear in response to a command:

*** ERROR - *message*
*** WARNING - *message.*

If an error occurs, the command is usually ignored. If only a warning is printed, the command is usually performed. If the message is not sufficiently informative, the appropriate command description may be helpful. The display after the command shows the effect of the command.

BLOT expects a valid database; a bad database can cause very unusual behavior. If a database format error is discovered before the time steps, the program prints an error of the following format:

DATABASE ERROR - Reading *database item*

and aborts. If a database format error is found while scanning the time steps, the following message is printed:

WARNING - End-of-file during time steps.

The time step with the error and all following time steps are ignored, but the program continues and the valid time steps are available for processing.

A random file is used to store the database variables for efficiency reasons. If the random file cannot be opened, a message of the following format is printed:

WARNING - Random file cannot be opened, *message*

and processing continues. Program performance may be impaired if the random file cannot be opened, but the program should produce correct results.

The program allocates memory dynamically as it is needed. If the system runs out of memory, the following message is printed:

FATAL ERROR - Too much dynamic memory requested

and the program aborts. The user should first try to obtain more memory on the system. Another solution is to run the program in a less memory-intensive fashion.

For example, cutting down the mesh size or requesting fewer curves in a plot set may require less memory.

BLOT has certain programmer-defined limitations (for example, the number of curves that may be defined). The limits are not specified in this manual since they may change. In most cases the limits are chosen to be more than adequate. If the user exceeds a limit, a message is printed. If the user feels the limit is too restrictive, the code sponsor should be notified so the limit may be raised in future releases of BLOT.

7. Executing BLOT

The details of executing BLOT are dependent on the system being used. The system manager of any system that runs BLOT should provide a supplement to this manual that explains how to run the program on that particular system. Site supplements for all currently supported systems are in Appendix F.

7.1 Execution Files

Table 7.1 summarizes the file usage of BLOT.

Description	Unit Number	Type	File Format
User input	standard input	input	Chapter 5
User output	standard output	output	ASCII
EXODUS database	11	input	Appendix A
Neutral file	20	optional output	Appendix B
Print file	21	optional output	ASCII
Log file	99	optional output	ASCII
Scratch random file	90	optional output	binary

Table 7.1. BLOT's file usage.

All files must be connected to the appropriate unit before a BLOT run. Each file (except standard input and output) is opened with the name retrieved by the EXNAME routine of the SUPES library [14].

The plots are generated on the connected graphics device(s). The graphics terminal device name (if any) should be assigned to symbol 1 (see EXNAME) and the hardcopy device name (if any) should be assigned to symbol 2. BLOT retrieves the device names with EXNAME. The appropriate graphic device drivers are linked to the program in a system-dependent fashion.

7.2 Special Software

BLOT is written in ANSI FORTRAN-77 [12] with the exception of the following system-dependent features:

- the VAX VMS help facility,
- the OPEN options for the files, and
- the use of ASCII characters that are not in the FORTRAN standard character set.

BLOT uses the following software packages:

- the Sandia Virtual Device Interface (SVDI) [10] with extensions for dual virtual devices (DVDI) [11] (if supported),
- the PLT graphics package [13] which uses the SVDI, and
- the SUPES package [14] which includes dynamic memory allocation, a free-field reader, and FORTRAN extensions.

References

[1] Amy P. Gilkey and Dennis P. Flanagan, "DETOUR – A Deformed Mesh / Contour Plot Program," SAND86-0914, Sandia National Laboratories, Albuquerque, NM, July 1987.

[2] Amy P. Gilkey, "TPLOT – A Time History or X-Y Plot Program for the Output of a Finite Element Analysis," SAND86-0883, Sandia National Laboratories, Albuquerque, NM, August 1986.

[3] Amy P. Gilkey, "SPLOT – A Distances-versus-variable Plot Program for the Output of a Finite Element Analysis," SAND86-0882, Sandia National Laboratories, Albuquerque, NM, August 1986.

[4] Amy P. Gilkey, "GROPE – An EXODUS Database Examination Program," Internal Memorandum RS1523/88/02, Sandia National Laboratories, March 1989.

[5] Zelma E. Beisinger and Charles M. Stone, "TPLOT2: A Flexible X-Y Plotting Program For Use With Finite Element Software," SAND80-2508, Sandia National Laboratories, Albuquerque, NM, February 1981.

[6] Mary R. Sagartz, "SPLOT: A Plot Program for Finite Element Data, Function vs. Distance," SAND84-1406, Sandia National Laboratories, Albuquerque, NM, January 1985.

[7] William C. Mills-Curran, Amy P. Gilkey, Dennis P. Flanagan, "EXODUS: A Finite Element File Format for Pre- and Post-Processing," SAND87-2997, Sandia National Laboratories, Albuquerque, NM, September 1988.

[8] Lee M. Taylor, Dennis P. Flanagan, and William C. Curran, "The GENESIS Finite Element Mesh File Format," SAND86-0910, Sandia National Laboratories, Albuquerque, NM, May 1986.

[9] C. R. Adams, "GRAFAID Code User Manual Version 2.0," SAND84-1725, Sandia National Laboratories, Albuquerque, NM, September 1985.

[10] Applied Computer Graphics Division, "Functional Specification of the Sandia Virtual Device Interface (SVDI)," SAND81-1900, Sandia National Laboratories, Albuquerque, NM, December 1981.

[11] Dennis P. Flanagan, "Dual Device, Shared Library, and Switchable Device Systems for the SVDI," SAND88-1341, Sandia National Laboratories, Albuquerque, NM, in preparation.

[12] *American National Standard Programming Language FORTRAN*, American National Standards Institute, ANSI X3.9-1978, New York, 1978.

[13] Colin B. Selleck and Mark P. Sears, "PLT – A Multipurpose Plotting Package," SAND83-0627, Sandia National Laboratories, Albuquerque, NM, August 1983.

[14] Dennis P. Flanagan, William C. Curran, and Lee M. Taylor, "SUPES – A Software Utility Package for Engineering Science," SAND86-0911, Sandia National Laboratories, Albuquerque, NM, September 1986.

A. The EXODUS Database Format

The following code segment reads an EXODUS database. The first segment is the GENESIS database format.

```
C   --Open the EXODUS database file

      NDB = 9
      OPEN (UNIT=NDB, ..., STATUS='OLD', FORM='UNFORMATTED')

C   --Read the title

      READ (NDB) TITLE
C       --TITLE - the title of the database (CHARACTER*80)

C   --Read the database sizing parameters

      READ (NDB) NUMNP, NDIM, NUMEL, NELBLK,
     &   NUMNPS, LNPSNL, NUMESS, LESSEL, LESSNL, NVERSN
C       --NUMNP - the number of nodes
C       --NDIM - the number of coordinates per node
C       --NUMEL - the number of elements
C       --NELBLK - the number of element blocks
C       --NUMNPS - the number of node sets
C       --LNPSNL - the length of the node sets node list
C       --NUMESS - the number of side sets
C       --LESSEL - the length of the side sets element list
C       --LESSNL - the length of the side sets node list
C       --NVERSN - the file format version number

C   --Read the nodal coordinates

      READ (NDB) ((CORD(INP,I), INP=1,NUMNP), I=1,NDIM)

C   --Read the element order map (each element must be listed once)

      READ (NDB) (MAPEL(IEL), IEL=1,NUMEL)
```

```
C   --Read the element blocks

      DO 100 IEB = 1, NELBLK

C      --Read the sizing parameters for this element block

         READ (NDB) IDELB, NUMELB, NUMLNK, NATRIB
C         --IDELB - the element block identification (must be unique)
C         --NUMELB - the number of elements in this block
C         --   (the sum of NUMELB for all blocks must equal NUMEL)
C         --NUMLNK - the number of nodes defining the connectivity
C         --   for an element in this block
C         --NATRIB - the number of element attributes for an element
C         --   in this block

C      --Read the connectivity for all elements in this block

         READ (NDB) ((LINK(J,I), J=1,NUMLNK, I=1,NUMELB)

C      --Read the attributes for all elements in this block

         READ (NDB) ((ATRIB(J,I), J=1,NATRIB, I=1,NUMELB)

  100 CONTINUE
```

```
C   --Read the node sets

      READ (NDB) (IDNPS(I), I=1,NUMNPS)
C      --IDNPS - the ID of each node set
      READ (NDB) (NNNPS(I), I=1,NUMNPS)
C      --NNNPS - the number of nodes in each node set
      READ (NDB) (IXNNPS(I), I=1,NUMNPS)
C      --IXNNPS - the index of the first node in each node set
C      --   (in LTNNPS and FACNPS)

      READ (NDB) (LTNNPS(I), I=1,LNPSNL)
C      --LTNNPS - the nodes in all the node sets
      READ (NDB) (FACNPS(I), I=1,LNPSNL)
C      --FACNPS - the factor for each node in LTNNPS

C   --Read the side sets

      READ (NDB) (IDESS(I), I=1,NUMESS)
C      --IDESS - the ID of each side set
      READ (NDB) (NEESS(I), I=1,NUMESS)
C      --NEESS - the number of elements in each side set
      READ (NDB) (NNESS(I), I=1,NUMESS)
C      --NNESS - the number of nodes in each side set
      READ (NDB) (IXEESS(I), I=1,NUMESS)
C      --IXEESS - the index of the first element in each side set
C      --   (in LTEESS)
      READ (NDB) (IXNESS(I), I=1,NUMESS)
C      --IXNESS - the index of the first node in each side set
C      --   (in LTNESS and FACESS)

      READ (NDB) (LTEESS(I), I=1,LESSEL)
C      --LTEESS - the elements in all the side sets
      READ (NDB) (LTNESS(I), I=1,LESSNL)
C      --LTNESS - the nodes in all the side sets
      READ (NDB) (FACESS(I), I=1,LESSNL)
C      --FACESS - the factor for each node in LTNESS
```

A valid GENESIS database may end at this point or at any point until the number of variables is read.

```
C   --Read the QA header information

      READ (NDB, END=900) NQAREC
C      --NQAREC - the number of QA records (must be at least 1)

      DO 110 IQA = 1, MAX(1,NQAREC)
         READ (NDB) (QATITL(I,IQA), I=1,4)
C         --QATITL - the QA title records; each record contains:
C         --    1) analysis code name (CHARACTER*8)
C         --    2) analysis code qa descriptor (CHARACTER*8)
C         --    3) analysis date (CHARACTER*8)
C         --    4) analysis time (CHARACTER*8)
  110 CONTINUE

C   --Read the optional header text

      READ (NDB, END=900) NINFO
C      --NINFO - the number of information records

      DO 120 I = 1, NINFO
         READ (NDB) INFO(I)
C         --INFO - extra information records (optional) that contain
C         --    any supportive documentation that the analysis code
C         --    developer wishes (CHARACTER*80)
  120 CONTINUE

C   --Read the coordinate names

      READ (NDB, END=900) (NAMECO(I), I=1,NDIM)
C      --NAMECO - the coordinate names (CHARACTER*8)

C   --Read the element type names

      READ (NDB, END=900) (NAMELB(I), I=1,NELBLK)
C      --NAMELB - the element type names (CHARACTER*8)
```

The GENESIS section of the database ends at this point.

```
C   --Read the history, global, nodal, and element variable information

      READ (NDB, END=900) NVARHI, NVARGL, NVARNP, NVAREL
C       --NVARHI - the number of history variables
C       --NVARGL - the number of global variables
C       --NVARNP - the number of nodal variables
C       --NVAREL - the number of element variables

      READ (NDB)
     &   (NAMEHV(I), I=1,NVARHI),
     &   (NAMEGV(I), I=1,NVARGL),
     &   (NAMENV(I), I=1,NVARNP),
     &   (NAMEEV(I), I=1,NVAREL)
C       --NAMEHI - the history variable names (CHARACTER*8)
C       --NAMEGV - the global variable names (CHARACTER*8)
C       --NAMENV - the nodal variable names (CHARACTER*8)
C       --NAMEEV - the element variable names (CHARACTER*8)

      READ (NDB) ((ISEVOK(I,J), I=1,NVAREL), J=1,NELBLK)
C       --ISEVOK - the name truth table for the element blocks;
C       --    ISEVOK(i,j) refers to variable i of element block j;
C       --    the value is 0 if and only if data will NOT be output for
C       --    variable i for element block j (otherwise the value is 1)
```

```
C   --Read the time steps

  130 CONTINUE
        READ (NDB, END=900) TIME, HISTFL
C         --TIME - the time step value
C         --HISTFL - the time step type flag:
C         --   0.0 for all variables output ("whole" time step) else
C         --   only history variables output ("history-only" time step)
C
        READ (NDB) (VALHV(IVAR), IVAR=1,NVARHI)
C         --VALHV - the history values for the current time step

        IF (HISTFL .EQ. 0.0) THEN
           READ (NDB) (VALGV(IVAR), IVAR=1,NVARGL)
C            --VALGV - the global values for the current time step

           DO 140 IVAR = 1, NVARNP
              READ (NDB) (VALNV(INP,IVAR), INP=1,NUMNP)
C               --VALNV - the nodal variables at each node
C               --   for the current time step
  140       CONTINUE

           DO 160 IBLK = 1, NELBLK
              DO 150 IVAR = 1, NVAREL
                 IF (ISEVOK(IVAR,IBLK) .NE. 0) THEN
                    READ (NDB) (VALEV(IEL,IVAR,IBLK),
     &                  IEL=1,NUMELB(IBLK))
C                    --VALEV - the element variables at each element
C                    --   for the current time step
                 END IF
  150         CONTINUE
  160       CONTINUE
        END IF

C      --Handle time step data
         ...
       GOTO 130

  900 CONTINUE
C   --Handle end of file on database
       ...
```

B. The GRAFAID Neutral File Format

The GRAFAID neutral file format is described in Section 9-2 of the GRAFAID Code User Manual [9]. The following description covers only those parts that are relevant to the BLOT program.

The neutral file is a sequential file written in ASCII format. Each record is free format with fields separated by a comma.

The general structure of the GRAFAID neutral file is:

Neutral file title record (optional)
1st Curve Definition Record Set
2nd Curve Definition Record Set
...
Nth Curve Definition Record Set
END NEUTRAL FILE

The neutral file title contains the name of the program which wrote the neutral file (BLOT) and the name, date, and time of the programs that created and last modified the database.

The format of the records in a curve definition record set is:

BEGIN CURVE,*curve-name*
ntitle,$title_1$
$title_2$
...
$title_{ntitle}$
x-label
y-label
x-min,*x-max*,*y-min*,*y-max*,*npts*,*aux-data*
axis-type,*x-type*,*aux-type*
x_1,y_1
x_2,y_2
...
x_{npts},y_{npts}
END CURVE,*curve-name*

Field	Type	Description
curve-name	15 characters	The curve name, set by the ACURVE and NCURVE commands.
ntitle	integer	The number of title lines (5 maximum).
$title_i$	80 characters	The i^{th} curve title line. If the user has requested QA information with the QA command, the database title is the first title line. The caption lines set by the CAPTION command are the next title lines. The last title line is a description of the curve.
x-label	40 characters	The X axis label, set by the XLABEL command.
y-label	40 characters	The Y axis label, set by the YLABEL command.
x-min	real	The X axis lower plot limit, set by the XSCALE command.
x-max	real	The X axis upper plot limit, set by the XSCALE command.
y-min	real	The Y axis lower plot limit, set by the YSCALE command.
y-max	real	The Y axis upper plot limit, set by the YSCALE command.
npts	integer	The number of curve data X-Y pairs.
aux-data	1 character	F to indicate that no auxiliary data exists.
axis-type	4 characters	NOLO to indicate a linear scale on both axes.
x-type	4 characters	NONM if the X data is nonmonotonic or MONO if the X data is monotonic.
aux-type	1 character	Blank field to indicate that no auxiliary data exists.
x_i	real	The i^{th} X data value.
y_i	real	The i^{th} Y data value.

C. Special Plot Text Capabilities

There are several special capabilities available for user-defined text that appears on a plot (such as the plot caption). For example, the text can include subscripted text, Greek letters, and special symbols. The user requests a capability by inserting an "escape sequence" in the text. The escape sequence starts with a backslash ("\") and (unless otherwise noted) must end with a space which is deleted from the text string. Escape sequences are translated only when **software** characters are plotted. If an invalid escape sequence is entered, an error message appears when the plot is drawn and the sequence is ignored. The following escape sequences are available:

\SQ	square	\CSQ	centered square
\DI	diamond	\CDI	centered diamond
\CS	cross	\CCS	centered cross
\X	X	\CX	centered X
\TR	triangle	\CTR	centered triangle
\CI	circle	\CCI	centered circle
\DO	dot	\CDO	centered dot
\LO	logo	\CLO	centered logo
\SLINE	solid line	\DLINE	dotted line
\DDLINE	dot-dash line	\SDLINE	short dash line
\LDLINE	long dash line	\MDLINE	medium dash line
\NEQ	not equal to		
\LEQ	less than or equal to	\GEQ	greater than or equal to
\NLEQ	not less than or equal to	\NGEQ	not greater than or equal to
\NLT	not less than	\NGT	not greater than
\LL	much less than	\GG	much greater than
\PLUSMIN	plus or minus	\PRIME	prime
\SUM	summation	\APPROX	approximation sign
\\	single backslash (with no ending space)		
\GR	Greek font	\ENG	normal font
\^	start superscript (with no ending space)		
_	start subscript (with no ending space)		

\- end superscript or subscript (with no ending space)

D. Command Summary

GENERAL COMMANDS

Program Control and Information (page 36)

LOG

requests that the log file be saved.

RESET

resets all plot parameters to their initial values.

EXIT

exits immediately from the program.

DETOUR

switches to the DETOUR subprogram.

PATHLINE

switches to the PATHLINE subprogram.

TPLOT

switches to the TPLOT subprogram.

SPLOT

switches to the SPLOT subprogram.

CMDFIL *file_name*

redirects BLOT input to the specified file.

SHOW *command*

displays plot parameters relevant to the command.

HELP *command*

displays information about a program command.

Database Listing and Printing (page 39)

SELECT *option*

selects database information for the LIST and PRINT commands.

LIST *option*

displays database information to the user's terminal.

PRINT *option*

prints database information to a file.

Time Step Selection (page 43)

TMIN *tmin*

sets the minimum selected time to *tmin.*

TMAX *tmax*

sets the maximum selected time to *tmax.*

NINTV *nintv*

sets the number of selected time intervals to *nintv* (delta offset).

ZINTV *nintv*

sets the number of selected time intervals to *nintv* (zero offset).

DELTIME *delt*

sets the selected time interval to *delt.*

ALLTIMES

selects all time steps between *tmin* and *tmax.*

TIMES [ADD,] t_1, t_2, ...

selects times t_1, t_2, etc.

STEPS [ADD,] n_1, n_2, ...

selects time steps n_1, n_2, etc.

HISTORY ON/OFF

controls whether history time steps are included in the selected time steps or only whole time steps.

QA ON/OFF. MESH/XY

controls whether the QA information is displayed on the plot legend.

AXIS ON/OFF, MESH/XY

controls whether the plot axes are numbered.

LEGEND ON/OFF, MESH/XY

controls whether the legend (excluding the QA information) is displayed on the plot legend.

CAPTION *line_number*, MESH/XY

sets the plot caption.

SOFTCHAR ON/OFF, *device*

selects software characters or hardware characters.

FONT STICK/SANSERIF/ROMAN, *device*

selects the font to use for software characters.

COLOR *ncol*

sets the maximum number of default color scale colors to use.

SPECTRUM *ncol*

sets the maximum number of spectrum colors to use.

BACKGROUND *color*

sets the background color for all plots.

FOREGROUND *color*

sets the foreground for all plots.

SNAP *nsnap*

controls the number of frames snapped on a camera device.

DISPVAR [ADD,] $variable_1$, $variable_2$, ...
DISPVAR ALL

DISPVAR ON/OFF

selects the variables whose values will be displayed on the plot legend.

Plot Set Display (page 50)

PLOT

generates the plots requested on the primary graphics device.

HARDCOPY

generates the plots requested on the hardcopy device.

AUTO ON/OFF

controls whether a response is requested after each plot.

Mesh View Control (page 52)

EMPTY

sets empty view.

DEFORM ON/OFF

controls whether the mesh is deformed or undeformed.

NUMBER NODES/ELEMENTS/ALL/OFF

controls whether the nodes and/or elements are numbered on the mesh.

MLINES ON/DOTTED/OFF

controls whether all mesh lines are displayed or only the mesh boundary and element block boundaries.

BOUNDARY ON/BLACK/OFF

controls whether the mesh boundary and element block boundaries are displayed on painted mesh plots.

NSETS [ADD,] set_id_1, set_id_2, ...
NSETS OFF

controls whether node sets are displayed on the mesh.

SSETS [ADD,] set_id_1, set_id_2, ...
SSETS OFF

controls whether side sets are displayed on the mesh.

Active Element Control (page 55)

BLOCKS [ADD or DELETE,] $block\_id_1$, $block\_id_2$, ...
BLOCKS OFF

selects the element blocks for active elements.

MATERIAL [ADD or DELETE,] $block\_id_1$, $block\_id_2$, ...
MATERIAL OFF

is equivalent to a **BLOCKS** command.

VISIBLE [ADD or DELETE,] *block_id*$_1$, *block_id*$_2$, ...

selects the element blocks that are visible in 3D.

DEATH ON, *element_variable*
DEATH OFF

controls whether "dead" elements are ignored.

DEADNODE ON/OFF

controls the display of "dead" nodes.

Multiple Views Control (page 57)

VIEW *nview*, *view_dependent_command*

sets the view-dependent parameters of view *nview*.

XVIEW ON/OFF

defines or deletes non-symmetric vertically divided views.

YVIEW ON/OFF

defines or deletes non-symmetric horizontally divided views.

XSYM LEFT/RIGHT, *xsym*
XSYM OFF

defines or deletes the symmetric vertically divided views.

YSYM BOTTOM/TOP, *ysym*
YSYM OFF

defines or deletes the symmetric horizontally divided views.

MULTTIME ON/OFF

assigns different time steps to each view or the same step to all views.

Mesh Control (page 60)

MAGNIFY *dfac*

causes the displacements to be magnified by *dfac* before they are added to the coordinates.

HIDDEN *level*

controls the 3D hidden line and surface algorithm.

ZOOM *hmin*, *hmax*, *vmin*, *vmax* or CURSOR
ZOOM TRANSLAT *hcen*, *vcen* or CURSOR
ZOOM EACH
ZOOM MESH

sets the mesh window limits.

SQUARE ON/OFF

determines if the mesh window is square or defined exactly by the ZOOM limits.

TICK *tick*

sets the axis tick interval for both mesh axes.

BLKCOL or MATCOL $color\_spec_1$, $color\_spec_2$, ...
BLKCOL or MATCOL RESET
BLKCOL or MATCOL ON/OFF

assigns colors to element blocks in wireframe and solid mesh plots.

LINETHICKNESS $line\_type_1$, $line\_thickness_1$, ...

assigns thicknesses to the lines that appear on mesh plots.

SPHERE ON/OFF

sets a flag specifying if all elements in the database are to be plotted as outlined circles.

FSPHERE ON/OFF

sets a flag specifying if all elements in the database are to be plotted as filled circles.

3D Rotation Commands (page 64)

ROTATE $axis_1$, $ndeg_1$, $axis_2$, $ndeg_2$, ...
ROTATE RESET

sets the 3D mesh rotation.

EYE *xpos*, *ypos*, *zpos*

sets the 3D mesh rotation by position.

CENTER *xcen*, *ycen*, *zcen*

sets the 3D center of rotation.

Mesh Display Mode Control (page 65)

WIREFRAM

sets wireframe mesh mode.

SOLID

sets solid mesh mode which paints the mesh.

CONTOUR *variable*

sets line contour mode which plots contours of the variable on the mesh.

PAINT *variable*

sets paint contour mode which paints contours of the variable on the mesh.

VECTOR *x_variable*, *y_variable*

sets vector mode which plots a vector representing the two variables on the mesh for each node or active element.

SYMBOL CRACK/ANGLE, *element_variable*

sets element symbol mode which plots a symbol on the mesh for each active element.

variable

is equivalent to a CONTOUR command if in wireframe or solid mesh mode; otherwise equivalent to the command which sets the current display mode.

Contour Control (page 68)

NCNTRS *ncntr*

sets the number of contours to *ncntr*.

CRANGE *cmin*, *cmax*

sets the minimum contour value to *cmin* and the maximum contour value to *cmax*.

DELCNTR *delc*, *cmin*

sets the contour interval to *delc* and the minimum contour value to *cmin*.

CSHIFT *cval*

sets the contour limits *cmin* and *cmax* so that *cval* falls on the nearest contour line or paint boundary.

Mode Specific Options (page 70)

CLABEL ON/*label_incr*/OFF

sets the contour labeling increment.

COPEN ON/OFF, ON/OFF

controls the plotting of values outside contour limits on a painted contour plot.

CSYMBOLS *nmin*, *nmax*

controls the plotting of symbols at the minimum and maximum nodal variable values on a contour plot.

VSCALE *vector_scale*

sets the vector/symbol scale factor.

PATHLINE COMMANDS

ADD

saves the pathlines defined in the last plot set.

REMOVE $pathline_1$, $pathline_2$, ... <no default>

deletes the specified pathlines or deletes the last -$pathline_i$ pathlines defined.

LOCATION *x_variable*, *y_variable*, *z_variable* (if 3D), <no default>$range_1$, $range_2$, ...

defines a pathline.

X-Y CURVE COMMANDS

Curve Definition (page 73)

ADD

saves the curves defined in the last plot set.

REMOVE $curve_1$, $curve_2$, ...<no default>

deletes the specified curves or deletes the last -$curve_i$ curves defined.

Axis Labeling and Scaling (page 74)

XLABEL

sets the X axis label.

YLABEL

sets the Y axis label.

XSCALE *xmin*, *xmax*

sets the X axis limits.

YSCALE *ymin*, *ymax*

sets the Y axis limits.

XTICK *xtick*

sets the X axis tick interval.

YTICK *ytick*

sets the Y axis tick interval.

SAMESCAL ON/OFF

controls whether all curves have the same scale.

NORMAL ON/OFF

controls whether each curve is to be scaled individually or all curves on a plot are to have the same scale.

RATIOXY *ratio*

sets the X-to-Y axis length ratio.

Curve Display Options (page 77)

OVERLAY VARIABLE/TIME/OFF

controls whether all curves are displayed on the same plot.

GRID ON/OFF

controls whether a grid is drawn on the plot.

LINES ON/VARY/OFF

controls the type of line to plot.

SYMBOLS VARY/*symbol_number*/OFF

controls whether symbols are drawn at each curve point.

CRVNUM FIRST/MIDDLE/LAST/OFF

controls where the curves are numbered.

Neutral File Options (page 79)

ACURVE *acurve*

sets the name part of the neutral file curve name.

NCURVE *ncurve*, *inccrv*

sets the starting curve number and increment for the neutral file.

NEUTRAL

writes the defined curves to the neutral file.

Mesh Display (page 81)

MESH

displays the mesh.

TPLOT COMMANDS

Curve Definition (page 82)

TYPLOT

defines a time history curve.

variable node_range$_1$ or *element_range*$_1$, *node_range*$_2$ or *element_range*$_2$, ...

is equivalent to a TYPLOT command.

XYPLOT

defines a variable-versus-variable curve.

SPLOT COMMANDS

Curve Definition (page 84)

NODES [ADD,] [PATH,] *node_range*$_1$ *node_range*$_2$ ...

selects the node numbers for the nodal variables.

ELEMENTS [ADD,] [PATH,] *element_range*$_1$ *element_range*$_2$ ...

selects the element numbers for the element variables.

SYPLOT

defines a distance-versus-variable curve.

variable

is equivalent to a SYPLOT command.

Mesh Display (page 85)

ECHO

displays the mesh with the nodes/elements selected with the NODES or ELEMENTS command.

E. Sample Plots

The plots on the following pages were generated with the sequence of BLOT commands that precede the plots. Note that the examples are from different databases.

Deformed Mesh, Contour, and Vector Plots

```
DETOUR                          $ Switch to DETOUR subprogram
QA OFF                          $ Turn off printing of QA information
TIME 2.104E-6                   $ Select database time
ZOOM 0, 5E-3, -1E-3, 4E-3       $ Set window limits
WIREFRAM                        $ Request a wireframe mesh plot (deformed)
PLOT                            $ Display the deformed wireframe mesh
                                $ Figure E.1

CONTOUR VONMISES                $ Request a contour plot of variable VONMISES
PLOT                            $ Display the contour plot
                                $ Figure E.2

VECTOR VELX, VELY               $ Request a vector plot of VELX and VELY
PLOT                            $ Display the vector plot
                                $ Figure E.3
```

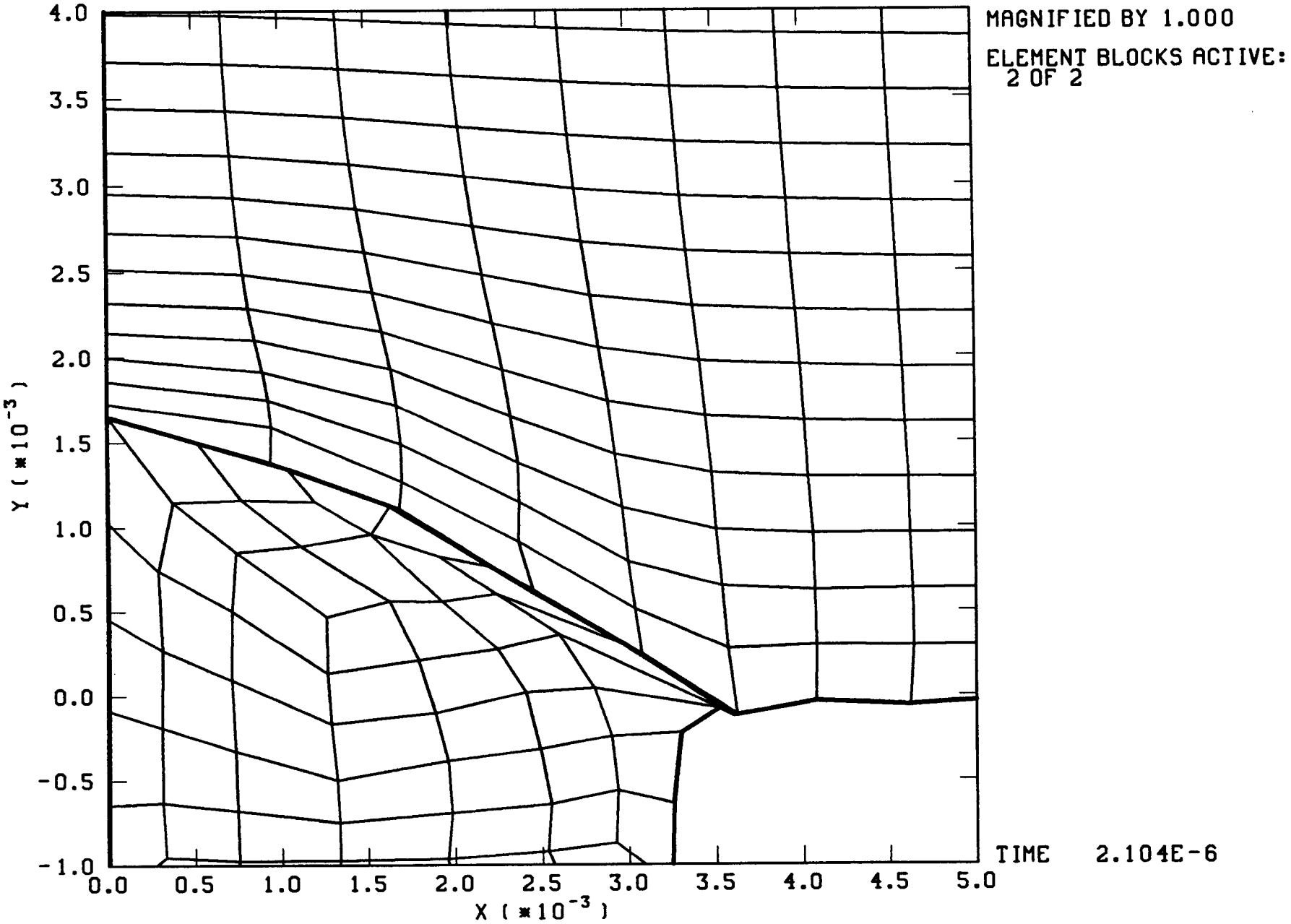

Figure E.1. Deformed Wireframe Mesh Plot

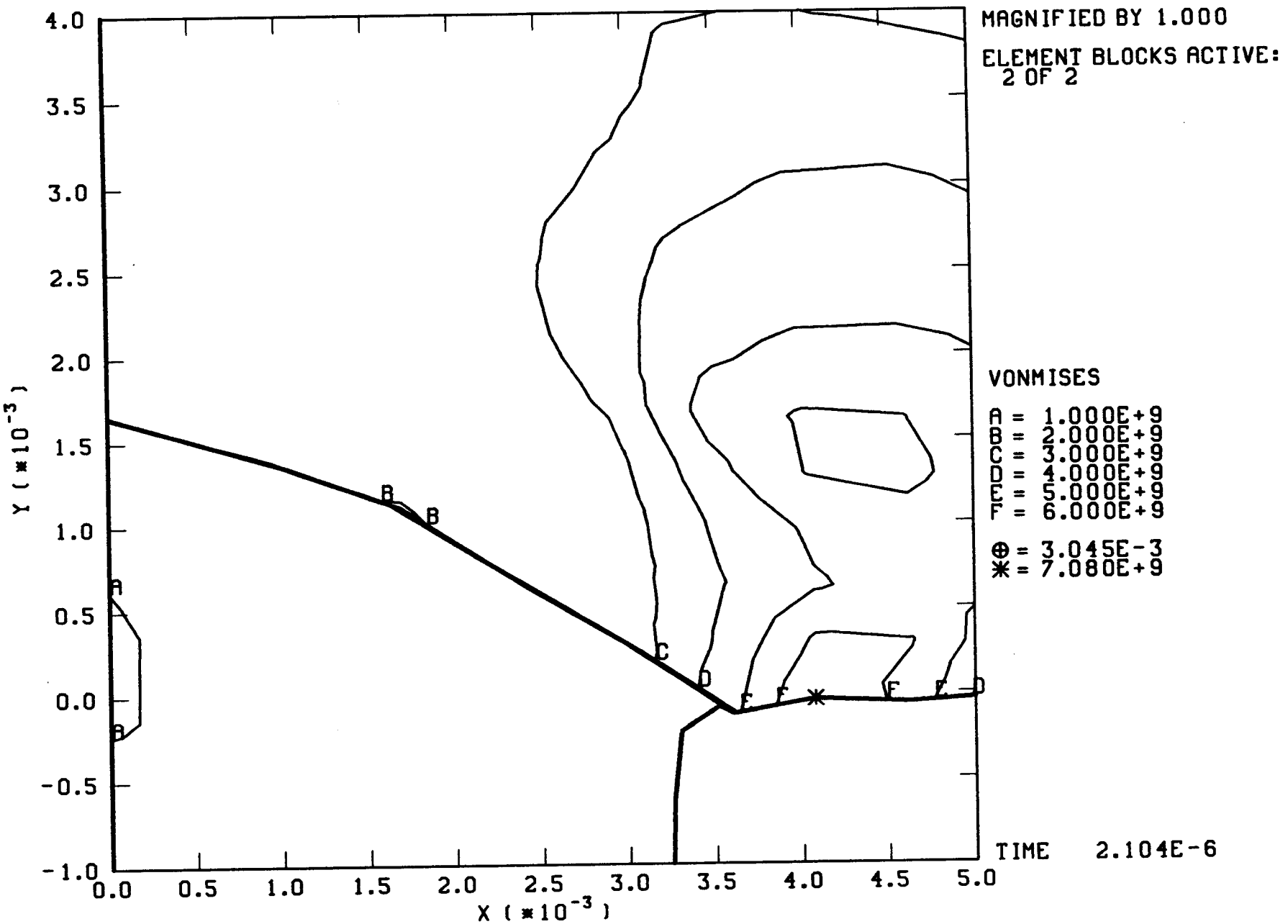

Figure E.2. Contour Plot of VonMises Stress

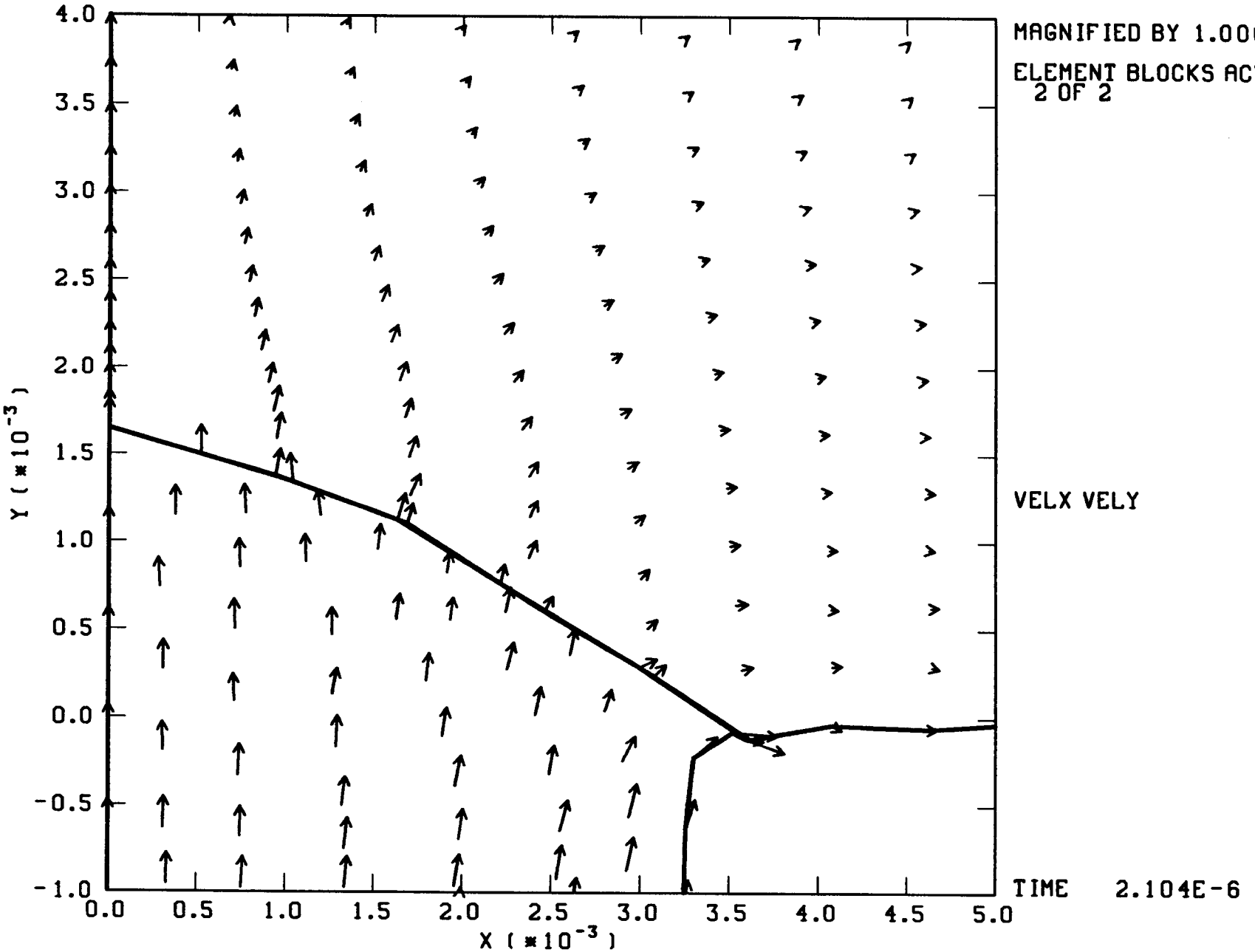

Figure E.3. Vector Plot of Velocities

Symmetric and Non-symmetric Views on a Mesh Plot

```
DETOUR                  $ Switch to DETOUR subprogram
QA OFF                  $ Turn off printing of QA information
TIME 12E-3              $ Select database time
WIREFRAM                $ Request a wireframe mesh plot
YVIEW                   $ Create non-symmetric views (bottom and top)
VIEW 2 DEFORM ON        $ Set top view to deformed
VIEW 4 DEFORM OFF       $ Set bottom view to undeformed
XSYM                    $ Create symmetric views (side by side)
PLOT                    $ Plot the views
                        $ Figure E.4
```

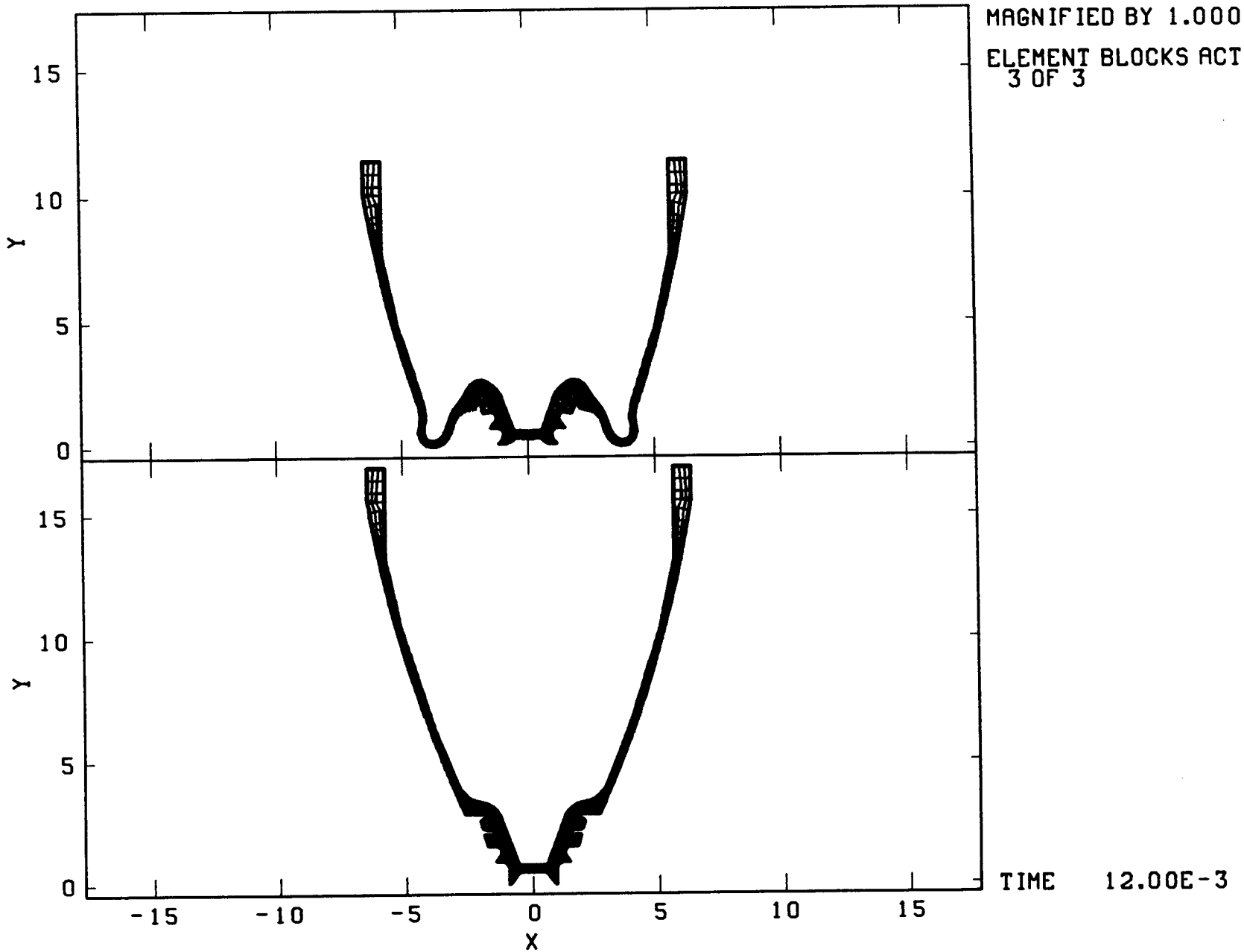

Figure E.4. Symmetric and Non-symmetric Views

Element Birth/Death Flags

```
DETOUR                  $ Switch to DETOUR subprogram
QA OFF                  $ Turn off printing of QA information
TIME .115E-3            $ Select database time
ZOOM 0,1.75,0,1.75      $ Set window limits
WIREFRAM                $ Request a wireframe mesh plot
XSYM LEFT 0.0           $ Create two symmetric views
YSYM BOTTOM 0.0         $ Create four symmetric views
PLOT                    $ Plot the views
                        $ Figure E.5

DEATH ON STATUS         $ Request element birth/death
PLOT                    $ Plot the views
                        $ Figure E.6
```

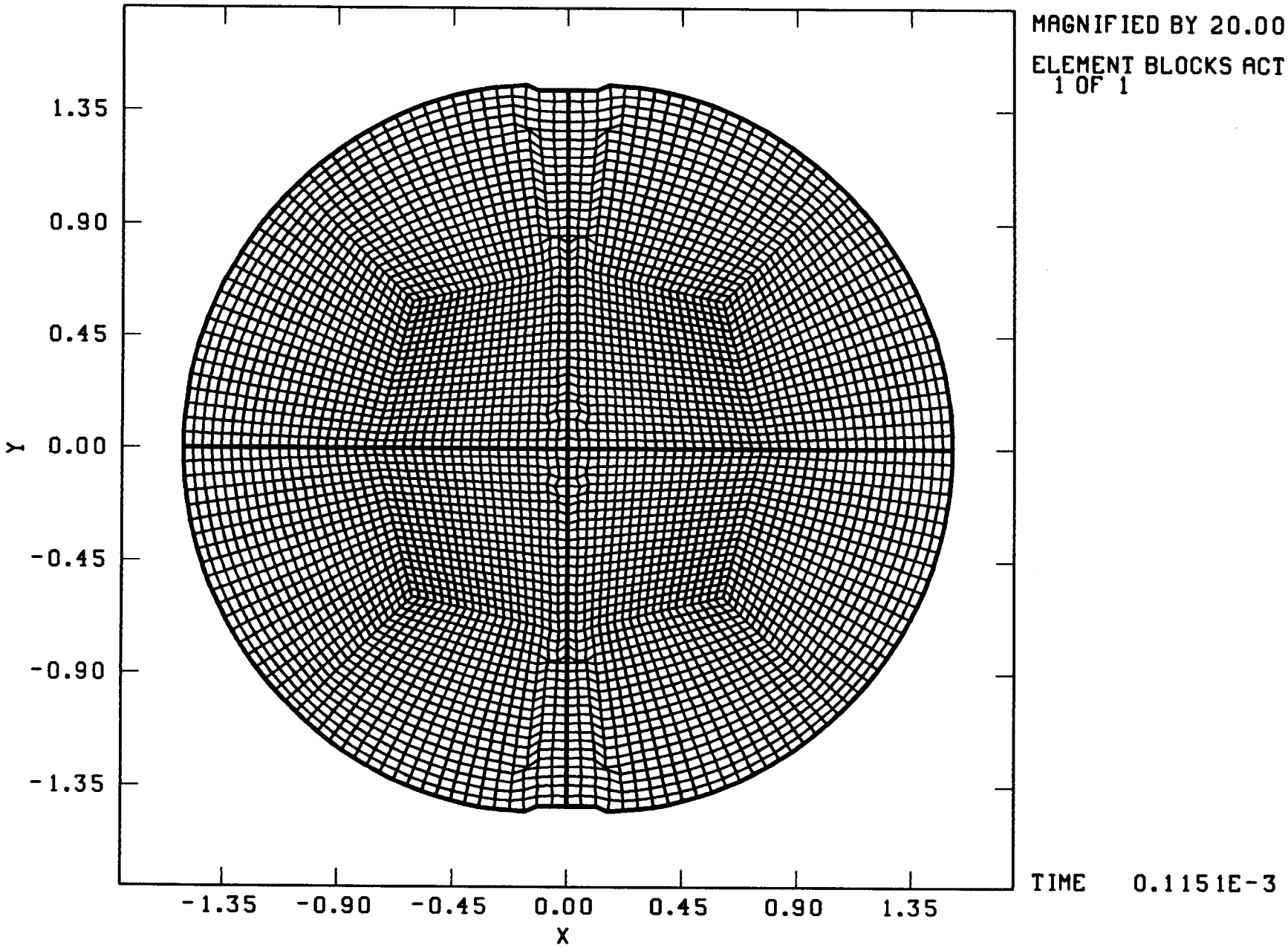

Figure E.5. Four Symmetric Views of Deformed Mesh with all Elements Selected

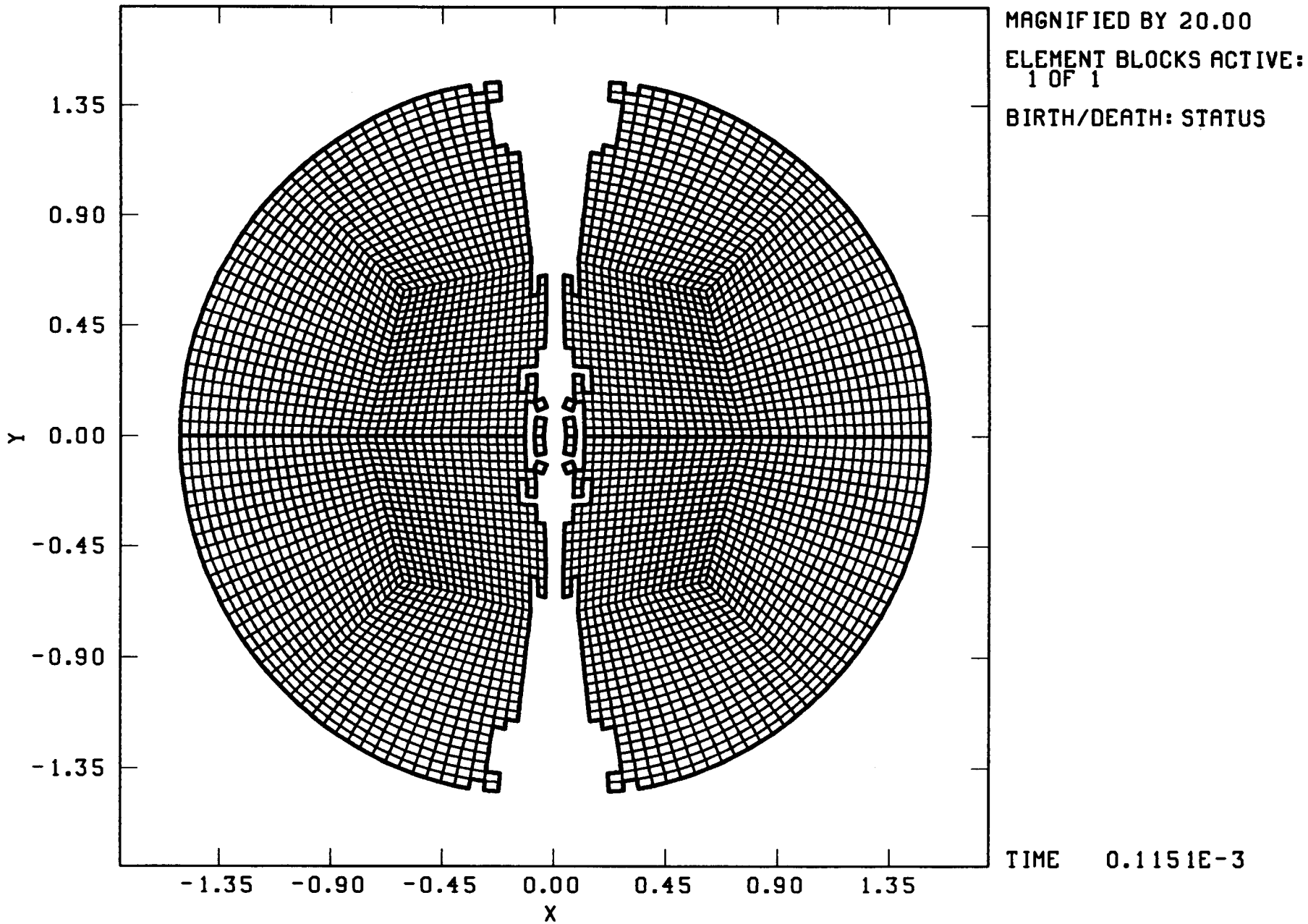

Figure E.6. Four Symmetric Views of Deformed Mesh with Element Birth/Death Flags Used

Node and Element Numbering

```
TPLOT                          $ Switch to TPLOT subprogram
NUMBER ALL                     $ Number both nodes and elements
ZOOM .000,.010,.000,.016       $ Zoom so numbers are readable
MESH                           $ Display single time step of mesh
                               $ Figure E.7
```

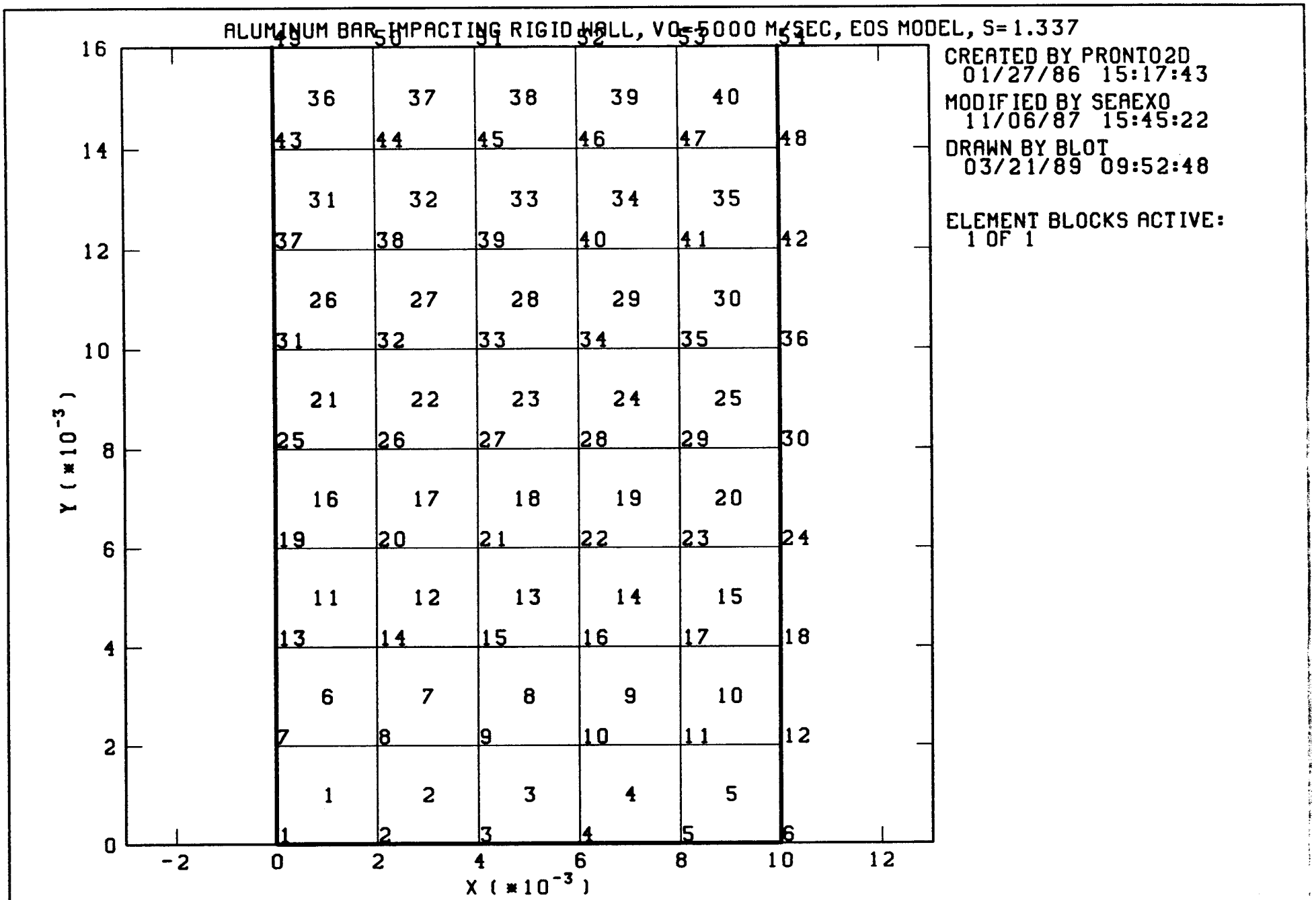

Figure E.7. Mesh Plot with Node and Element Numbering

Time History Plot

```
TPLOT                    $ Switch to TPLOT subprogram
QA OFF                   $ Turn off printing of QA information
SIGXX 50 TO 250 BY 50    $ Define 5 time history curves
                         $       at elements 50,100,150,200,250
OVERLAY                  $ Overlay all curves on one plot
LINE VARY                $ Draw each curve with a different line type
PLOT                     $ Plot the curves defined above
                         $ Figure E.8
```

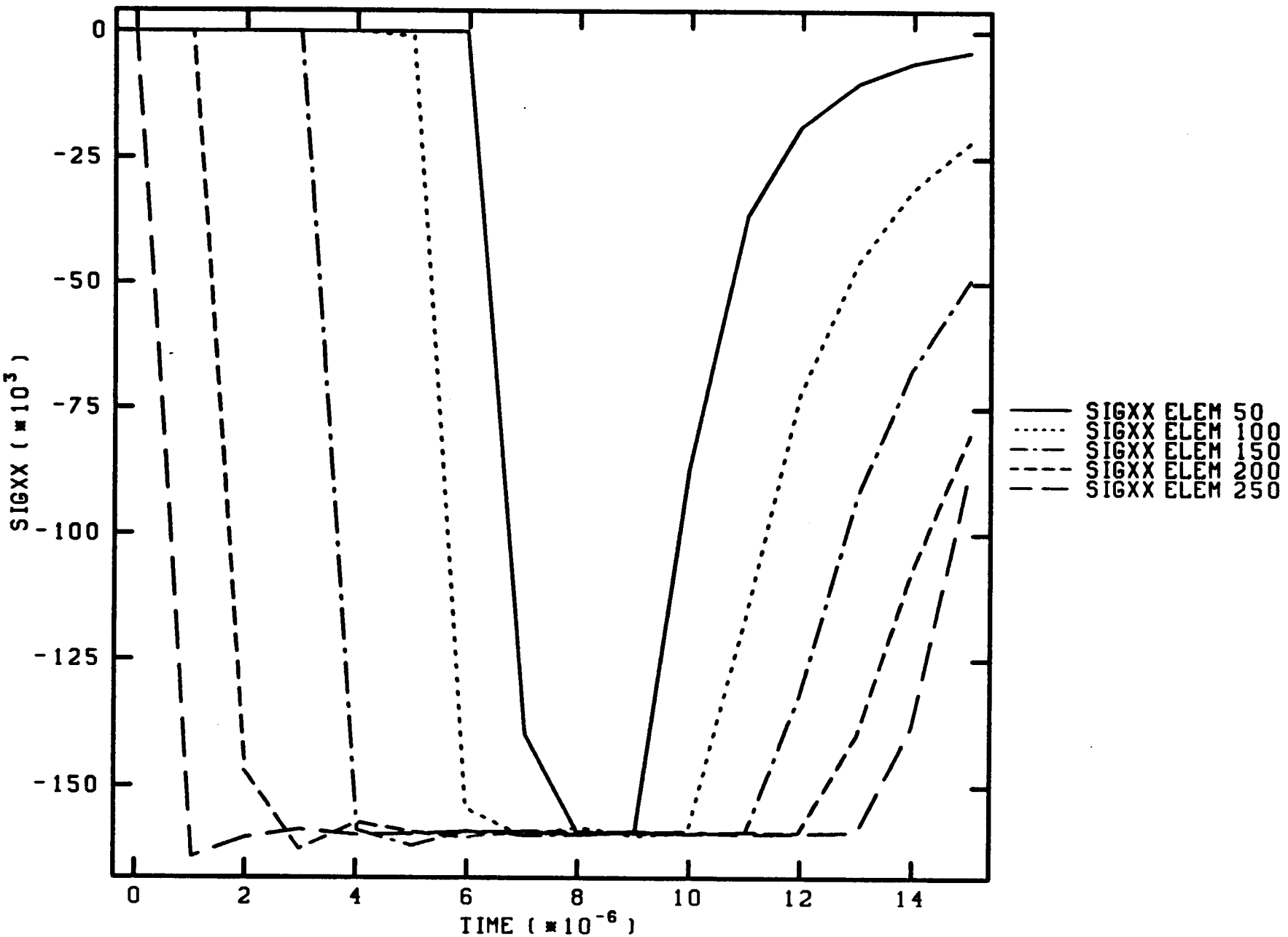

Figure E.8. Time History Plot

Variable-versus-variable Plot

```
TPLOT                    $ Switch to TPLOT subprogram
XYPLOT                   $ Define a variable-versus-variable curve
STRAIN,2                 $       X variable: STRAIN at element 2
VONMISES,2               $       Y variable: VONMISES at element 2
GRID                     $ Draw grid
SYMBOLS                  $ Draw symbols at each point on curve
XSCALE -.03,.45          $ Set X axis limits
YSCALE -50,800           $ Set Y axis limits
YLABEL                   $ Define Y axis label
VONMISES STRESS
PLOT                     $ Plot the curve defined above
                         $ Figure E.9
```

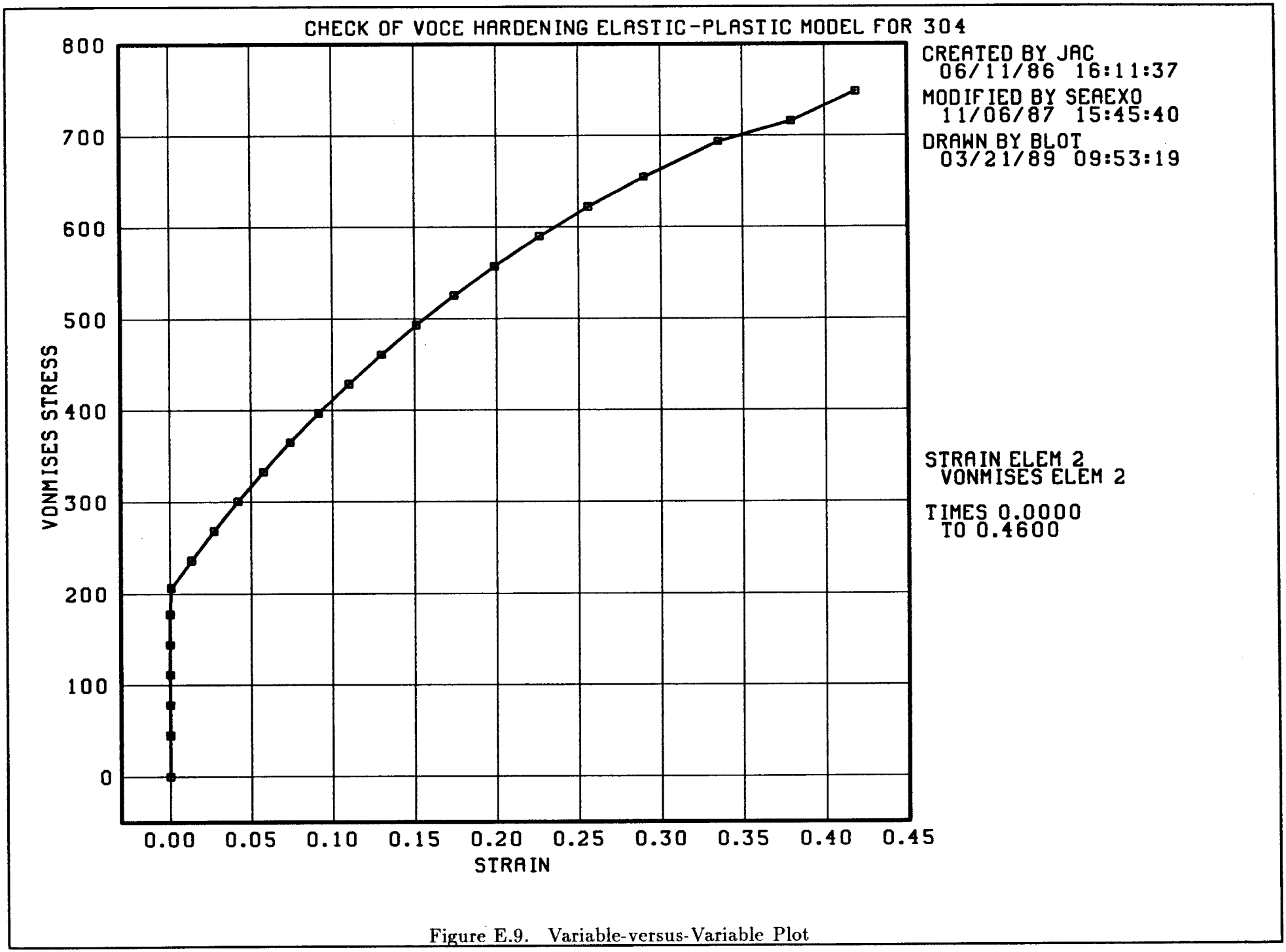

Figure E.9. Variable-versus-Variable Plot

Distance-verus-Variable Plots

```
SPLOT                       $ Switch to SPLOT subprogram
NODES 1 TO 306 BY 6         $ Select nodes 1,7,13,...,301
ACCLX                       $ Define curve of nodal variable ACCLX
ACCLY                       $ Define curve of nodal variable ACCLY
TIMES 8.0E-6                $ Select time 8E-6 only
OVERLAY VARIABLE            $ Overlay curves for both variables on one plot
SYMBOLS                     $ Draw symbols at each point on curve
YSCALE -37E9,5E9            $ Set Y axis limits
PLOT                        $ Plot the curves defined above
                            $ Figure E.10
```

```
RESET                       $ Clear all parameters (for example, symbols)
ELEMENTS 3 TO 250 BY 5      $ Select elements 3,8,13,...,248
SYPLOT SIGXX                $ Define curve of element variable SIGXX
NINTV 6                     $ Select 6 times over the database
OVERLAY TIME                $ Overlay curves for all times on one plot
LINE VARY                   $ Draw each curve with a different line type
PLOT                        $ Plot the curves defined above
                            $ Figure E.11
```

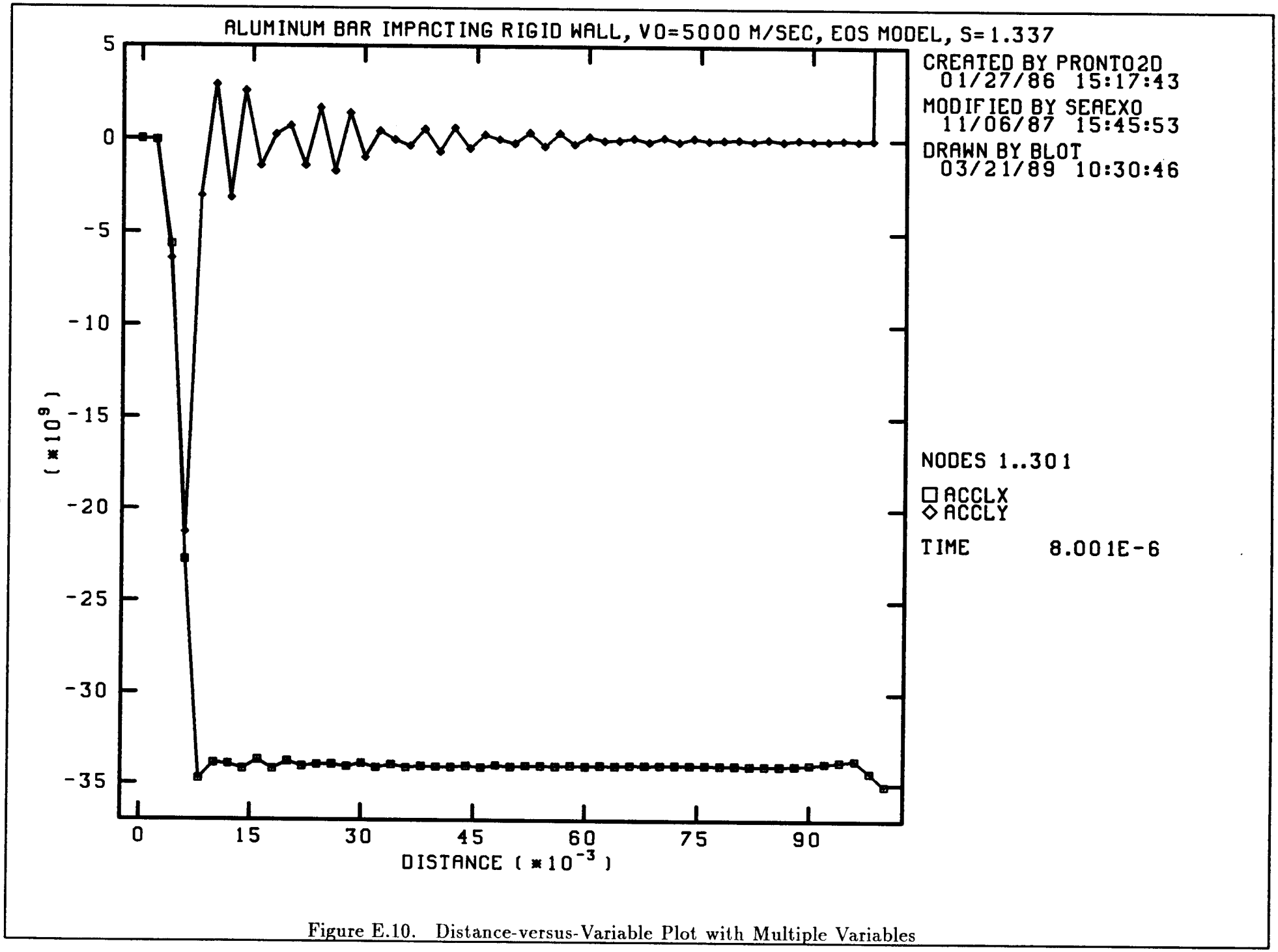

Figure E.10. Distance-versus-Variable Plot with Multiple Variables

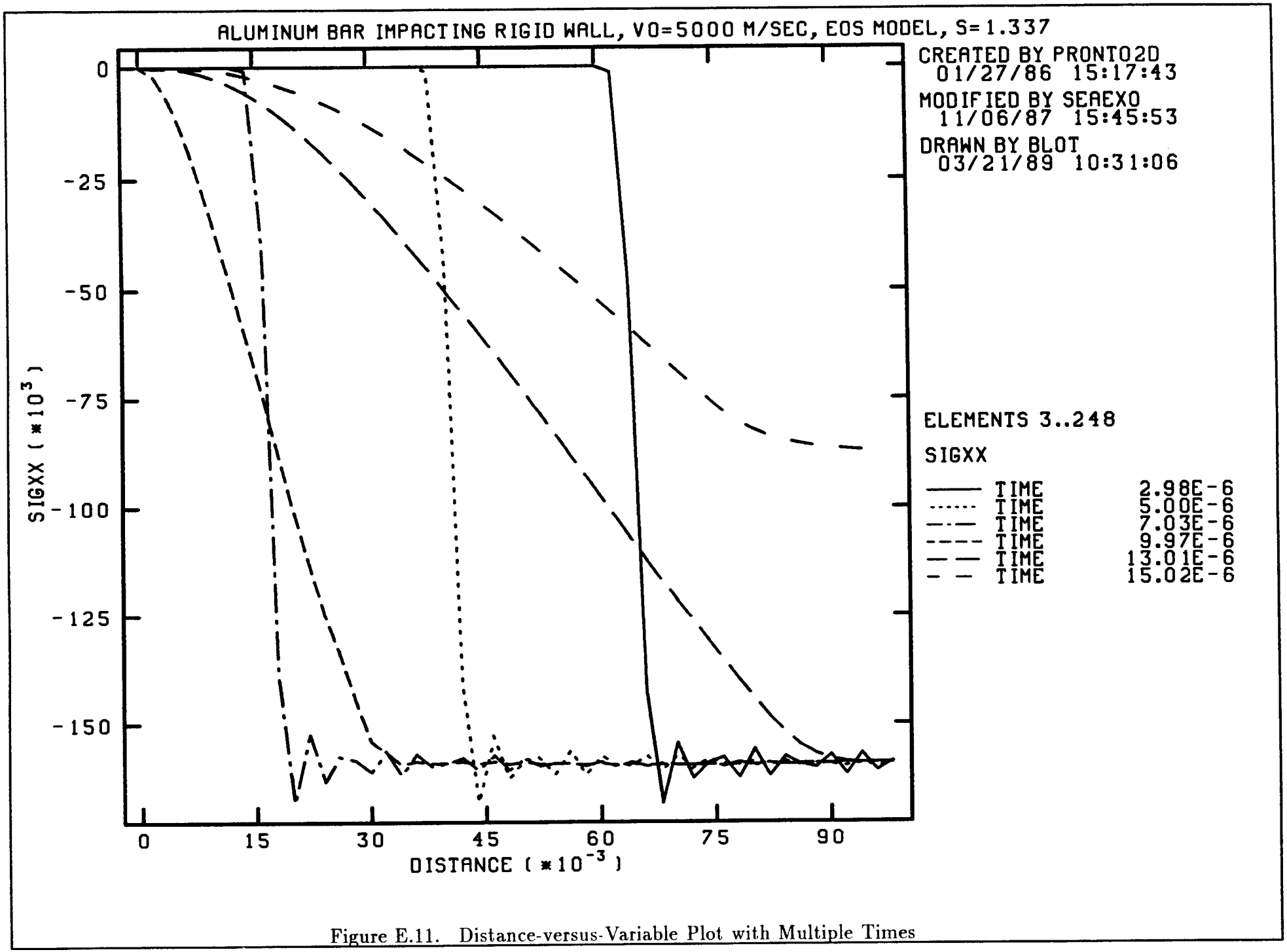

Figure E.11. Distance-versus-Variable Plot with Multiple Times

3D Model Rotation and Node Set Display

```
DETOUR                  $ Switch to DETOUR subprogram
VISIBLE 1               $ Make only element block 1 visible
PLOT                    $ Plot the mesh
                        $ Figure E.12

QA OFF                  $ Turn off printing of QA information
ROTATE X 23.0 y -34.    $ Rotate the model 23 degrees around the X axis and
                        $ -34 degrees around the Y axis
PLOT                    $ Plot the mesh
                        $ Figure E.13

NSETS 1                 $ Mark the nodes defined in node set 1
PLOT                    $ Plot the mesh
                        $ Figure E.14
```

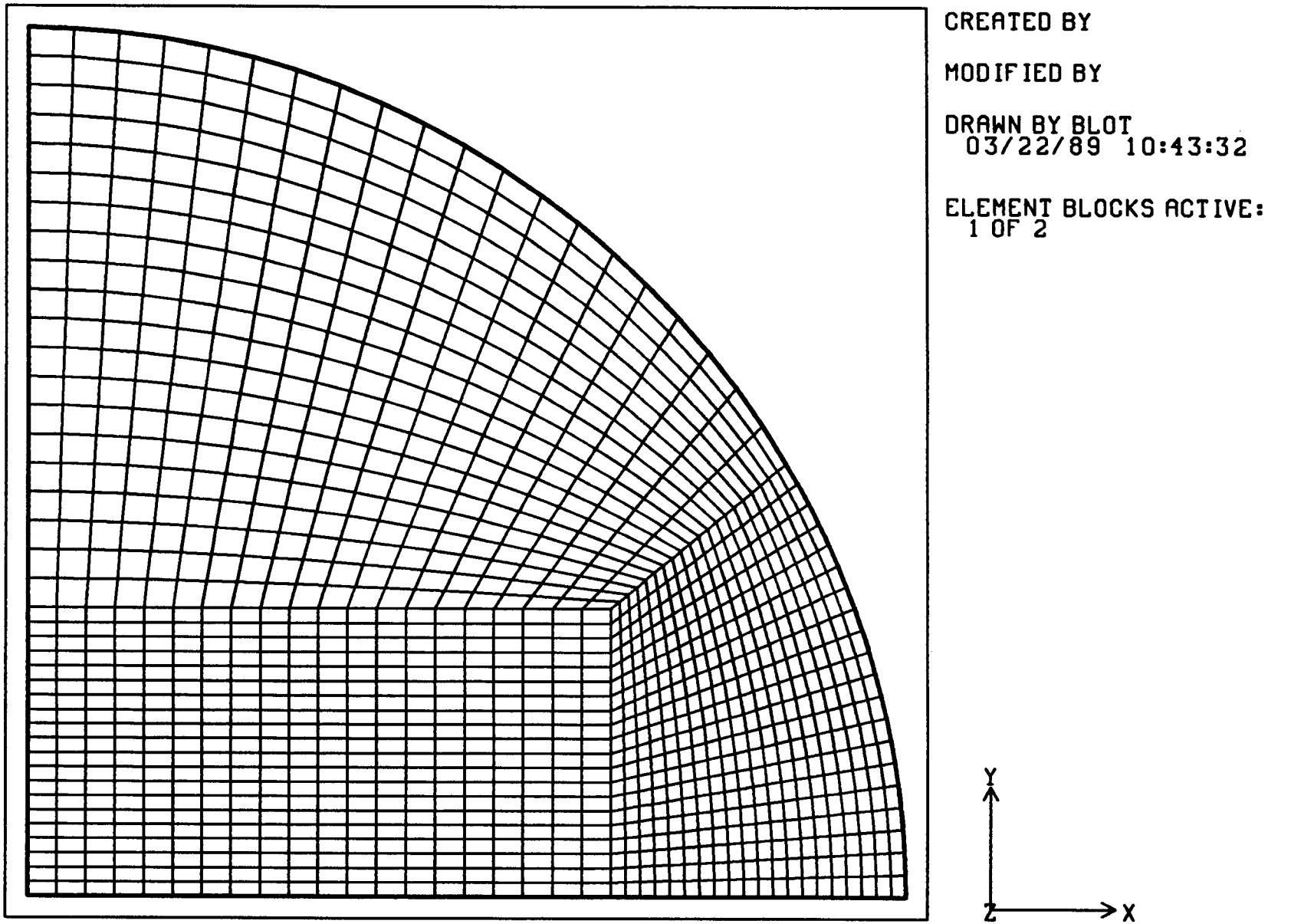

Figure E.12. 3D Model Viewed Along the Z Axis

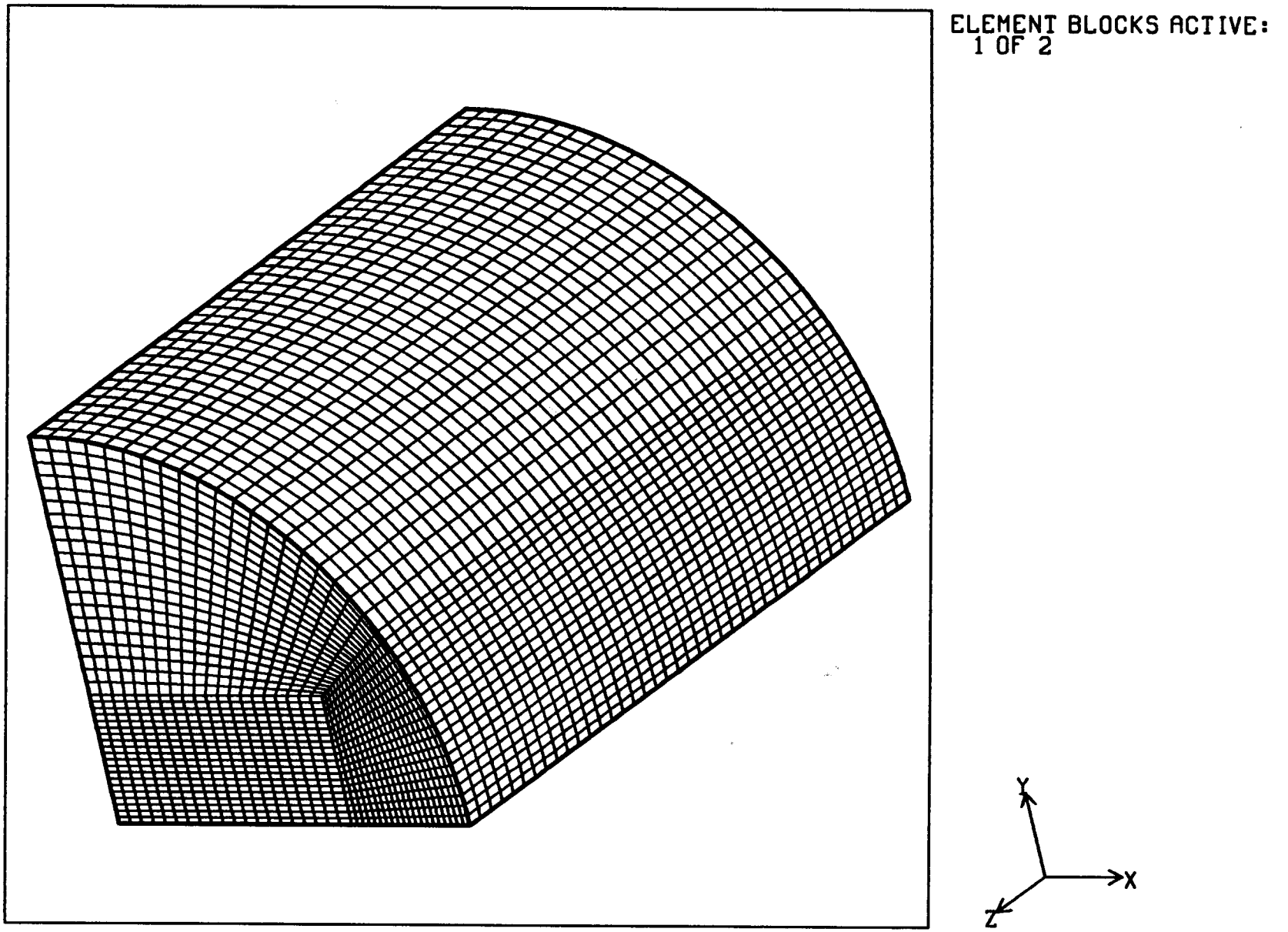

Figure E.13. Rotated 3D Model

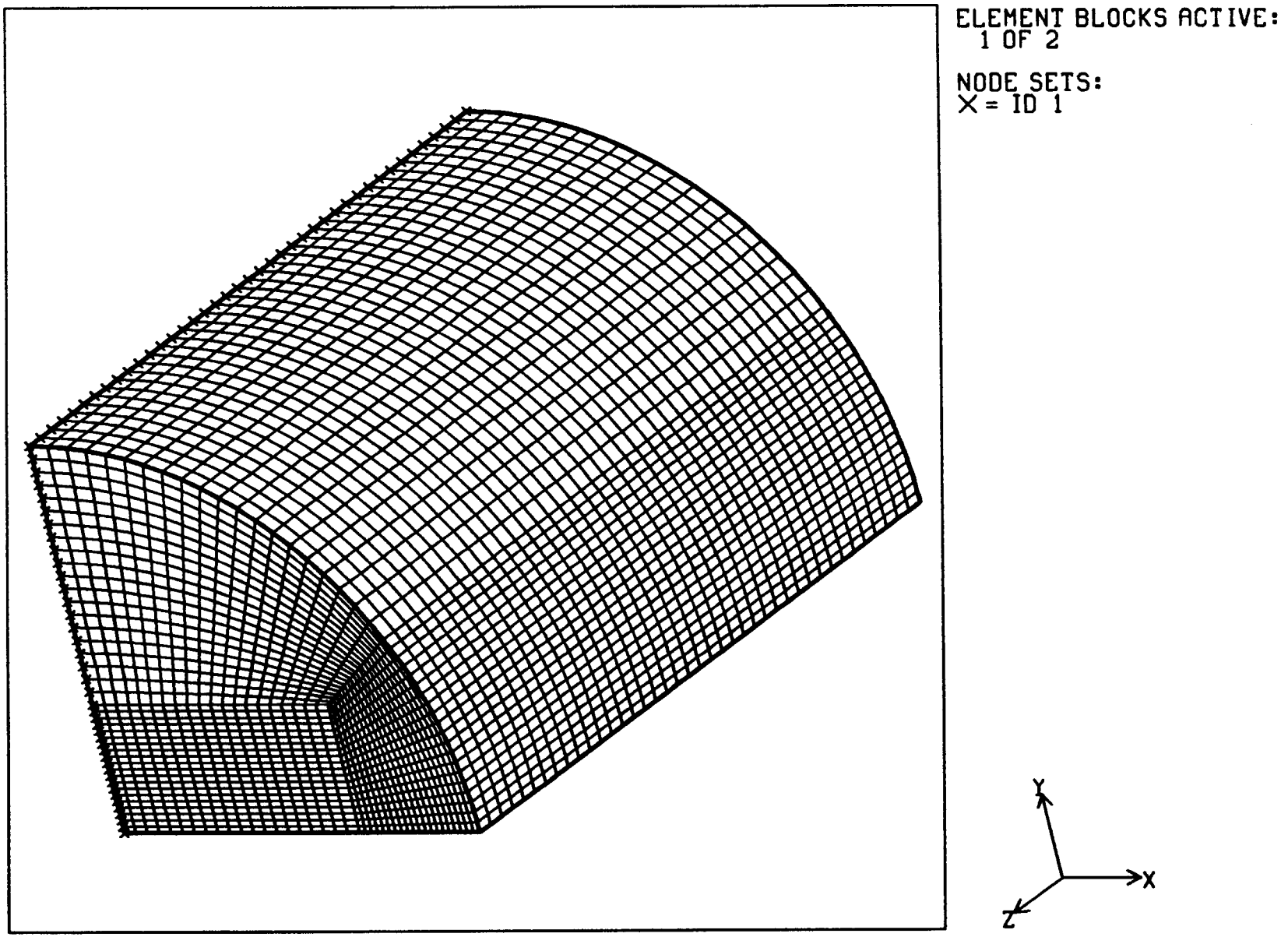

Figure E.14. Rotated 3D Model With Node Set 1 Marked

Mesh and Contour Plot of Rotated 3D Model

```
DETOUR                    $ Switch to DETOUR subprogram
ROTATE X 30 Y -30         $ Rotate the model 30 degrees around the X axis and
                          $ -30 degrees around the Y axis
XVIEW ON                  $ Define two vertically divided views
                          $ By default, both will be wireframe mesh plots
VIEW 2 CONTOUR DISY       $ Define the second view to be a contour plot
                          $ of the nodal variable DISY
PLOT                      $ Plot the mesh and contours
                          $ Figure E.15
```

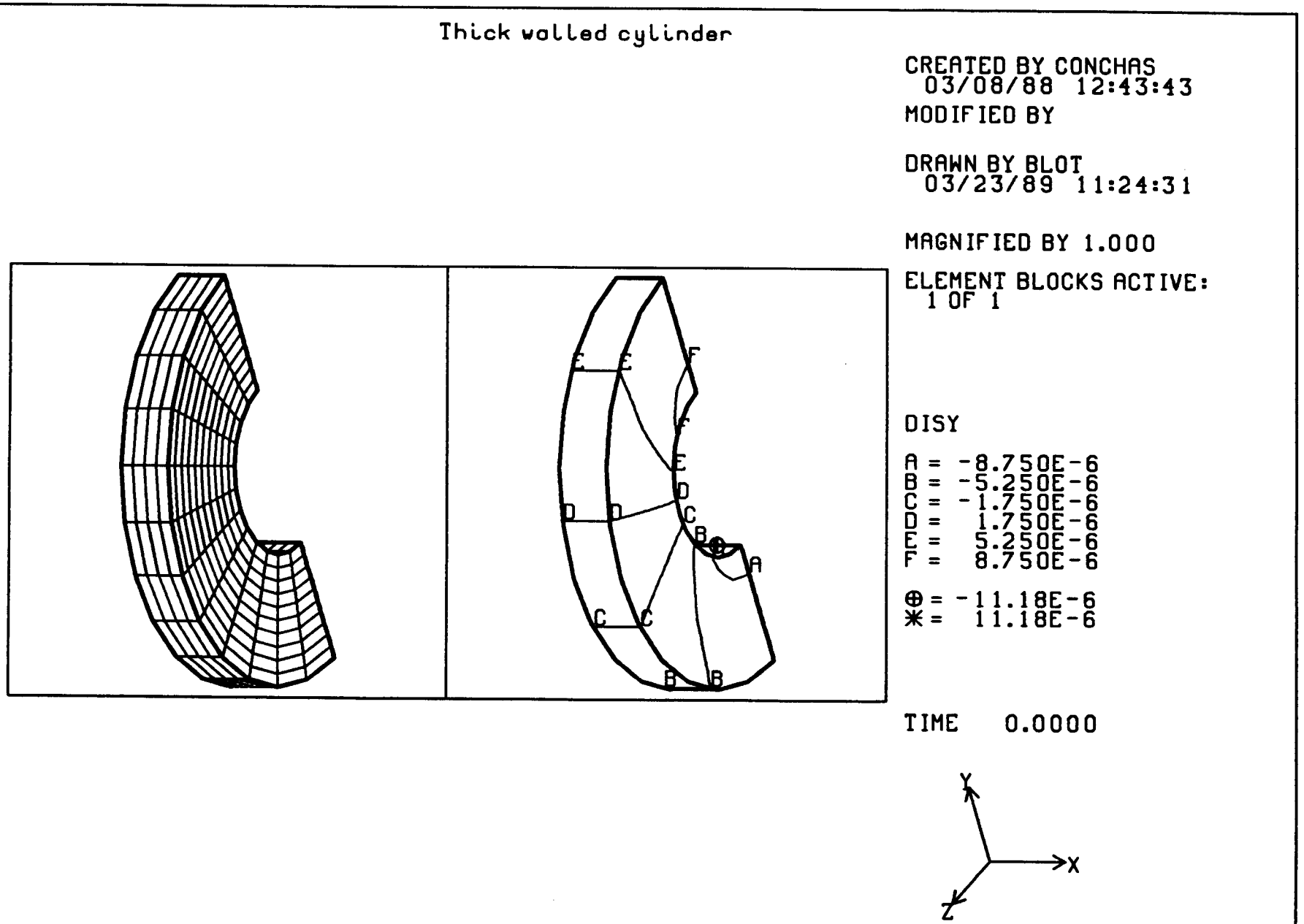

Figure E.15. Mesh and Contour Plot of Rotated 3D Model

Pathline Plot

```
QA OFF                              $ Turn off printing of QA information
PATHLINE                            $ Switch to PATHLINE subprogram
ZOOM -3000 8000 -13000 -2000        $ Set window limits
LOCATION X_COORD Y_COORD Z_COORD    $ Specify the variables that define the coordinates
                                    $ of the points on the pathline
PLOT                                $ Plot the pathline
                                    $ Figure E.16
```

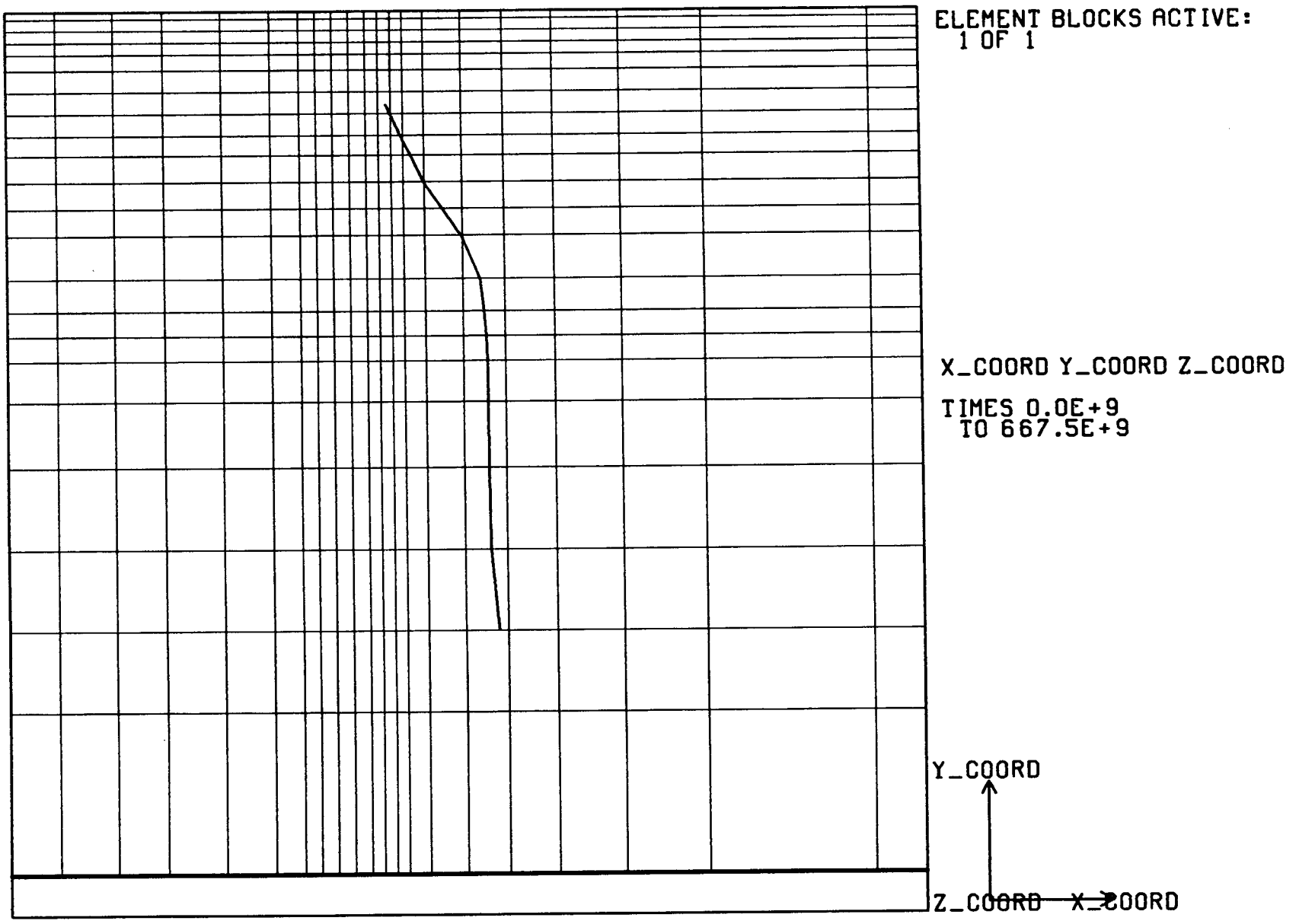

Figure E.16. Pathline Plot

F. Site Supplements

F.1 VAX VMS

The command to execute BLOT on VMS is:

BLOT *database primary_device hardcopy_device user_input batch_flag*

Database is the filename of the input EXODUS database. A prompt appears if *database* is omitted. The default is TAPE11.EXO.

Primary_device is the VDI device code of the primary graphics device. A prompt listing the available devices appears if *primary_device* is omitted. The default is the last selected device.

Hardcopy_device is the VDI device code of the hardcopy device. A prompt listing the available devices appears only if both *primary_device* and *hardcopy_device* are omitted. The default is no hardcopy device.

If *user_input* is given, the user input is read from this file. If BLOT is executed as a batch job, this file must be specified. If BLOT is executed interactively and *user_input* does not end with an EXIT command, the user is prompted for further BLOT commands. If BLOT is executed interactively and *user_input* is not specified, the user is prompted for BLOT commands.

If *batch_flag* is specified as b, the BLOT session is submitted as a batch job. Omitting this parameter executes BLOT interactively.

F.2 CRAY CTSS

To execute BLOT, the user must have selected the **acclib** library and be running **ccl**.

The command to execute BLOT on CTSS is:

blot *database device* i=*input* o=*output*

Database is the filename of the input EXODUS database. The default is **tape11**.

Device is the VDI device code of the graphics device. Only a single device is allowed on this system. The default is LS5.

User input is read from *input*, which defaults to **tty** (the terminal keyword). User output is directed to *output*, which defaults to **tty** (the terminal).

Distribution:

1510 J. W. Nunziato
1511 D. K. Gartling
1511 R. R. Eaton
1511 J. H. Glick (30)
1511 P. L. Hopkins
1512 J. C. Cummings
1513 D. W. Larson
1520 L. W. Davison
1521 R. D. Krieg and Staff (11)
1522 R. C. Reuter, Jr. and Staff (15)
1523 J. H. Biffle and Staff (13)
1524 L. W. Davison, Actg. and Staff (11)
1530 D B. Hayes
1531 S. L. Thompson
1533 S. T. Montgomery
1533 A. C. Robinson
1533 P. Yarrington
1550 C. W. Peterson, Jr.
1556 W. L. Oberkampf
2814 P. F. Chavez
3141 S. A. Landenberger (5)
3151 W. I. Klein (3)
3141-1 for DOE/OSTI (8)
5213 J. K. Deuel
6258 B. J. Thorne
6258 D. S. Preece
6314 L. S. Costin
6322 R. E. Glass
6334 H. J. Iuzzolino
6334 R. D. McCurley
6334 J. S. Rath
6334 R. P. Rechard
6334 E. Shepherd
6452 J. D. Miller
6514 R. D. Watson
8240 C. W. Robinson
8241 G. A. Benedetti
8242 M. R. Birnbaum
8243 M. L. Callabresi
8244 C. M. Hartwig
8245 R. J. Kee
8524 J. A. Wackerly

Distribution:

1510 J. C. Cummings
1511 D. K. Gartling
1511 R. R. Eaton
1511 P. L. Hopkins
1513 D. W. Larson
1520 L. W. Davison
1521 H. L. Morgan and Staff (12)
1522 R. C. Reuter, Jr. and Staff (14)
1523 J. H. Biffle and Staff (12)
1524 D. R. Martinez and Staff (15)
1530 J. R. Asay
1531 E. S. Hertel
1531 J. M. McGlaun
1531 A. C. Robinson
1531 S. L. Thompson
1533 P. Yarrington
1533 S. T. Montgomery
1534 P. L. Stanton
1550 C. W. Peterson, Jr.
1556 W. L. Oberkampf
2814 P. F. Chavez
3141 S. A. Landenberger (5)
5213 J. K. Deuel
6258 B. J. Thorne
6258 D. S. Preece
6314 L. S. Costin
6322 R. E. Glass
6342 H. J. Iuzzolino
6342 R. D. McCurley
6342 R. P. Rechard
6452 J. R. Miller
8240 C. W. Robinson
8241 G. A. Benedetti
8242 M. R. Birnbaum
8243 M. L. Callabresi
8244 C. M. Hartwig
8245 R. J. Kee
8524 J. A. Wackerly

Sandia National Laboratories
Albuquerque, New Mexico 87185

date: August 21, 1990

to: Distribution

from: Ray J. Meyers, 1521

subject: Updates to the postprocessing program BLOT

1 Introduction

This memo describes recent changes that have been made to the BLOT program since the BLOT manual [1] was published. The changes include:

- New DETOUR Commands
  - Model Slicing
  - New Zoom Commands
  - Axis-Only Plotting
- Bugs Fixed
  - QA and Info Record Handling
  - Incorrect Graph Information
  - Compatibility With New SUPES
  - New Default Color Scheme
- X11 Driver
  - Patched Spectrum Color Problem

2 DETOUR Commands

2.1 Model Slicing

The capability for slicing through models has been finished and documented. The algorithm as implemented checks each element of the mesh, plotting any element which is at least partially visible with respect to the defined cutting plane. Because individual elements remain whole, the algorithm gives the best result when cutting parallel to element faces. Several forms of the CUT command are explained below.

CUT *x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4*

CUT CURSOR

This command expects three x, y, z coordinates defining a plane in space, and a fourth point on the side of the plane which is to be kept after slicing. The CURSOR form of the command prompts the user to input (using the cursor) three nodes whose coordinates are used to define the cutting plane, and a fourth node on the side of the plane to be kept after cutting.

CUT NORM *x1, y1, z1, x_norm, y_norm, z_norm*

CUT NORM CURSOR

This command expects a single point located on the desired cutting plane, and a normal direction which defines the normal of the plane. This normal vector need not be normalized on input. The CURSOR form of the command prompts the user to input (using the cursor) a node to be located on the cutting plane, and a second node whose direction with respect to the first node defines a normal vector for the plane.

CUT SCREEN

This command prompts the user to define with the cursor two points on the screen for the desired cutting plane. The third point on the cutting plane is assumed to be the position of the observer's eye. The user also selects a node on the side of the cutting plane to be kept when slicing. The effect of this command is to slice the model using a plane perpendicular to the screen.

CUT OFF

This command disables the CUT option.

Known Bugs. Compound cuts are not allowed on the model. If the user issues a second CUT command on a currently cut model, the resulting plot may occasionally be incorrect. The only sure way to avoid this is to issue a CUT OFF command and redraw the uncut model, and then select a cutting plane using the uncut model.

2.2 New Zoom Commands

New ZOOM commands have been added to zoom in on a fixed point in space or to track a specific node.

ZOOM RADIUS *x, y, radius* (2-D)

ZOOM RADIUS *x, y, z, radius* (3-D)

ZOOM RADIUS sets the zoom window to be centered on the given coordinate and holds it there through subsequent transformations of the model. Any point within the distance *radius* of the center coordinate will remain within the ZOOM window.

ZOOM NODE *node_id, radius*

ZOOM NODE CURSOR

ZOOM NODE sets the zoom window to be centered on the specified node and tracks the node through transformations and displacements. All points within the distance *radius* of the node will remain within the ZOOM window. The CURSOR form of the command prompts the user to select (using the cursor) the node to be centered. A second node is selected with the distance between the first and second nodes to be used as the *radius* of the ZOOM window.

2.3 Axis-Only plotting

A command has been added to allow the user to plot only the legend and orientation axis for a plot, saving time in orienting large three-dimensional models.

PLOT AXIS

PLOT AXIS plots only the legend and orientation axis for the current plot. It is not a toggle, and the AXIS qualifier must be used each time an axis only plot is needed.

3 Bugs Fixed

QA and Info Record Handling. BLOT defines parameters for maximum number of QA records (MAXQA=25) and Info records (MAXINF=25) which can be handled by the program. For input files which contain more records than the maximum, an error occurred. For example, for an input file of 30 records, BLOT would read the first 25 records, but try to write 30 records, which would output garbage to the terminal. The new version has been modified to read the *last* 25 recordsfrom the file, and write at most 25 records to the terminal.

Incorrect Graph Information. In TPLOT and SPLOT, incorrect information about variables was echoed to the screen. For instance, a plot with stress for the y-axis function and strain for the x-axis function was described incorrectly as a plot of 'strain vs. stress'. This has been corrected to echo the correct 'stress vs. strain' information.

Compatibility With New SUPES. The new version of BLOT has been changed to conform to the new release of SUPES. This may require minor changes in script files

for anyone attempting to run BLOT in the master-slave mode on the Cray. If there are problems, contact Ray Meyers or John Red-Horse.

New Default Color Scheme. The default color scheme for the contour and paint options of DETOUR has been changed. The new scheme uses the same six colors, but attempts to order them in a more intuitive manner from colder (blue) to warmer (red).

4 X11 Driver

The X11 driver has been patched to correct the problem allocating colors in the SPECTRUM command of BLOT . In order to correct the problem temporarily, the number of colors allowed for the driver has been reduced from 256 to 128. This does not adversely affect BLOT inasmuch as anything above 48 colors appears essentially smooth. Dino Pavlakos is currently working to provide a permanent fix for this problem, and to enhance the driver to provide a backing store for the window so that the contents are not lost each time it is obscured.

Known Bugs. If many different numbers of colors are specified in repeated SPECTRUM commands, it is possible to run out of colors, resulting in a color spectrum which is no longer pleasing. It is not easy to do this, but if you should, the only fix currently is to exit the program and start again.

5 Future Enhancements

Most of the modifications documented above are in reponse to suggestions received from users. If you submitted a request which has not been acted upon, please contact Ray Meyers to find the current status of the request. Attached to this memo is a form to submit future requests. Comments on the new commands documented here would be especially appreciated.

References

[1] Amy P. Gilkey, John H. Glick, "BLOT - A Mesh and Curve Plot Program for the Output of a Finite Element Analysis," SAND88-1432, Sandia National Laboratories, Albuquerque, NM, June, 1989.

enc: blot request form

BLOT ENHANCEMENT / BUG FIX REQUEST

NAME:_____ DATE:_____

ORGANIZATION:_____PHONE:_____ VERSION OF BLOT:_____

COMPUTING ENVIRONMENT (eg., CTSS on CRAY; VMS on Vaxcluster, Vaxstation 3100):

PRIORITY: _____ _____ _____ _____
High Medium Low Back burner

RETURN FORM TO: L.A. Schoof, org 1511 or R.J Meyers, org. 1523 at 4-3905

ENHANCEMENT

Fill in this section if you are requesting an enhancement.

Brief description of enhancement; include what is currently done; attach additional sheets of diagrams, plots, etc.

Suggested approach:

BUG REPORT

Fill in this section if you have encountered a bug.

Brief description of bug; attach additional sheets containing plots which show the bug; include machine error messages, if applicable.

Command sequence which caused the bug.